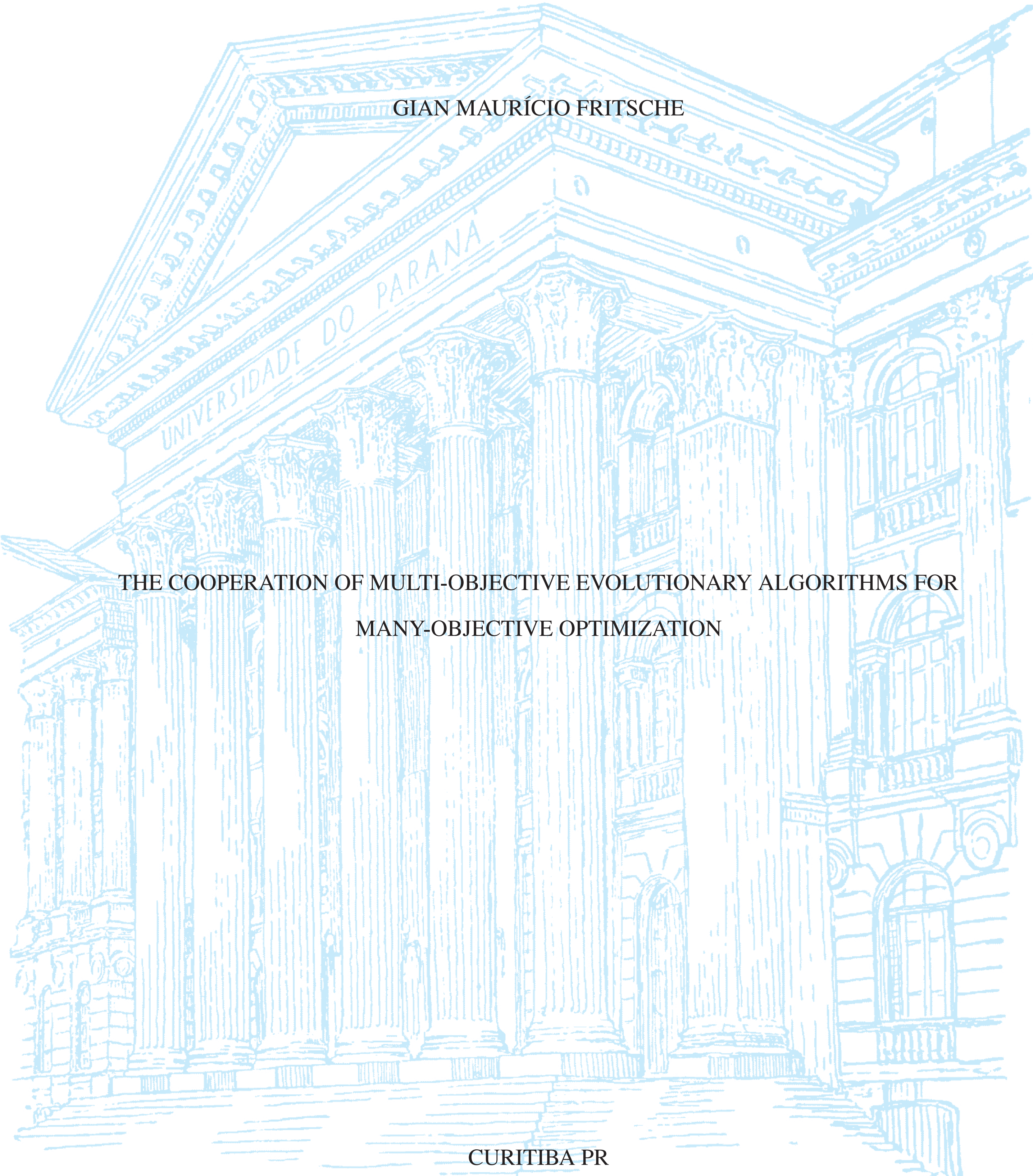
UNIVERSIDADE FEDERAL DO PARANÁ



GIAN MAURÍCIO FRITSCHE

THE COOPERATION OF MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS FOR MANY-OBJECTIVE OPTIMIZATION

CURITIBA PR

2020

GIAN MAURÍCIO FRITSCHE

THE COOPERATION OF MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS FOR MANY-OBJECTIVE OPTIMIZATION

Tese apresentada como requisito parcial à obtenção do grau de Doutor em Ciência da Computação no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Aurora Trinidad Ramirez Pozo.

CURITIBA PR

2020

Catalogação na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

F919c Fritsche, Gian Maurício
The cooperation of multi-objective evolutionary algorithms for many-objective optimization [recurso eletrônico] / Gian Maurício Fritsche. – Curitiba, 2020.

Tese - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática, 2020.

Orientador: Aurora Trinidad Ramirez Pozo

1. Algoritmos. 2. Otimização combinatória. 3. Programação heurística. I. Universidade Federal do Paraná. II. Pozo, Aurora Trinidad Ramirez. III. Título.

CDD: 511.8

Bibliotecário: Elias Barbosa da Silva CRB-9/1894

MINISTÉRIO DA EDUCAÇÃO
SETOR DE CIENCIAS EXATAS
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO INFORMÁTICA - 40001016034P5

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **GIAN MAURICIO FRITSCHE** intitulada: **THE COOPERATION OF MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS FOR MANY-OBJECTIVE OPTIMIZATION**, sob orientação da Profa. Dra. AURORA TRINIDAD RAMIREZ POZO, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de doutor está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 14 de Dezembro de 2020.

Assinatura Eletrônica
15/12/2020 11:31:02.0
AURORA TRINIDAD RAMIREZ POZO
Presidente da Banca Examinadora

Assinatura Eletrônica
15/12/2020 11:34:50.0
ANDRÉ BRITTO DE CARVALHO
Avaliador Externo (UNIVERSIDADE FEDERAL DE SERGIPE)

Assinatura Eletrônica
17/12/2020 08:00:54.0
RICHARD ADERBAL GONÇALVES
Avaliador Externo (UNIVERSIDADE ESTADUAL DO CENTRO-OESTE)

Assinatura Eletrônica
15/12/2020 16:57:42.0
WAGNER MACHADO NUNAN ZOLA
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
18/12/2020 07:03:56.0
CAROLINA PAULA DE ALMEIDA
Avaliador Externo (UNIVERSIDADE ESTADUAL DO CENTRO-OESTE)

To my wife.

ACKNOWLEDGEMENTS

First, I would like to thank God for all the blessings and amazing people in my life. Also, thanks a lot to my wife, Giovanna Garcia Fritsche, for all support and encouragement. Thanks to my mother, Carmen Fritsche, for the encouragement and financial assistance at the beginning of this journey. I also thank my advisor, Prof. Dr. Aurora Trinidad Ramirez Pozo, for her valuable insights and guidance. Finally, thank all my friends for help and support, especially from the Cbio/GrES lab and the Lord Riders Motorcycle Club. Also, I acknowledge CAPES and CAPES-DS for partial financial support.

RESUMO

A otimização simultânea de múltiplos objetivos está presente em várias instâncias de problemas para diferentes campos de pesquisa. Algoritmos Evolutivos Multiobjetivo (*Multi-objective Evolutionary Algorithm* — MOEA) têm sido amplamente aplicados para resolver esses problemas. No entanto, quando o número de objetivos é maior que três, chamados de problemas com muitos objetivos (*Many-objective Optimization Problems* — MaOP), esses problemas apresentam desafios para os MOEAs. Por exemplo, entre esses desafios está a chamada “falta de pressão de seleção”, pois a maioria das soluções se tornam não-comparáveis. Nos últimos anos, pesquisadores têm investigado e proposto diversos algoritmos para a otimização com muitos objetivos. No entanto, dadas as características das diferentes instâncias de problema e as características dos diferentes algoritmos, nenhum algoritmo é melhor do que todos os outros para todos os problemas, conhecido como o teorema “sem almoço grátis”, (do inglês *No Free Lunch*). A pesquisa explorada nesta tese foca na execução cooperativa de diferentes MOEAs, considerando um conjunto de problemas com uma diversidade de características. Nossa hipótese é que a cooperação pode ser eficaz para mais problemas do que cada MOEA aplicado isoladamente. Com base nesta hipótese, um modelo distribuído para cooperação entre MOEAs foi proposto. Nesse modelo, diferentes MOEAs são executados, trocando informações. O modelo proposto foi avaliado inicialmente usando comunicação síncrona e assíncrona para a cooperação de dois MOEAs. Os resultados foram favoráveis à versão síncrona, que favorece a troca de informações. Os resultados alcançados motivaram a continuidade da linha de investigação proposta. Após análise experimental, concluímos que a participação dos algoritmos no modelo cooperativo pode ser ponderada para priorizar aqueles que apresentam melhores resultados. Para isso, avaliamos a incorporação de hiper-heurística para guiar a busca. Além disso, usamos a etapa de migração proposta para trocar informações entre o MOEA executado e os demais. Avaliamos o uso de hiper-heurísticas com e sem a etapa de migração proposta. A versão com troca de informações obteve melhores resultados. Os resultados alcançados também foram competitivos ao melhor MOEA para a maioria das instâncias de problemas, com uma melhor avaliação geral. Desta forma, demonstramos a importância do método de migração proposto para incorporar informações externas no processo evolutivo de cada algoritmo. Em seguida, desenvolvemos uma versão aprimorada, combinando conhecimentos de várias validações e análises experimentais. Essa versão foi então comparada a uma hiper-heurística estado da arte para otimização multiobjetivo. Os resultados foram favoráveis à abordagem cooperativa proposta. Finalmente, avaliamos o modelo proposto em um problema do mundo real. Este problema otimiza os parâmetros de construção de uma turbina eólica. Os resultados alcançados são comparáveis aos do melhor MOEA avaliado para este problema. Concluímos que o método de troca de informações proposto é uma abordagem eficaz para a cooperação de vários MOEAs para otimização com muitos objetivos. Além disso, concluímos que esta pode alcançar resultados competitivos para uma ampla variedade de instâncias de problemas.

Palavras-chave: Otimização com muitos objetivos, Hiper-heurística, Otimização com variáveis contínuas, Seleção de heurísticas, Algoritmos Evolutivos, Problema do mundo real

ABSTRACT

The simultaneous optimization of multiple objectives is present in several problem instances for different research fields. Multi-objective Evolutionary Algorithms (MOEA) have been widely applied to solve these problems. However, when the number of objectives is higher than three, these problems, called many-objective problems (MaOP), pose challenges for MOEAs. For example, among these challenges is the so-called "lack of selection pressure", as most solutions become not-comparable. In recent years, researchers have investigated and proposed several algorithms for many-objective optimization. However, given the characteristics of the different problem instances and the characteristics of the different algorithms, no one algorithm is better than all others for all problems, known as the No Free Lunch theorem. The research explored in this thesis focuses on the cooperative execution of different MOEAs, considering a set of problems with a diversity of characteristics. Our hypothesis is that the cooperation may be effective for more problems than each MOEA applied standalone. Based on this hypothesis, a distributed model for MOEAs cooperation was proposed. In this model, different MOEAs are executed, exchanging information. The proposed model was initially evaluated using synchronous and asynchronous communication for the cooperation of two MOEAs. The results were favorable to the synchronous version, which favors the exchange of information. The results achieved motivated the continuity of the proposed line of investigation. After experimental analysis, we concluded that the participation of the algorithms in the cooperative model could be weighed to prioritize those that present better results. For that, we evaluated the incorporation of hyper-heuristics to guide the search. Also, we use the proposed migration step to exchange information between the executed MOEA and the others. We evaluated the use of hyper-heuristics with and without the proposed migration approach. The version with the exchange of information obtained better results. The results achieved were also competitive to the best MOEA for most problem instances, with better overall results. In this way, we demonstrate the importance of the proposed migration method to incorporate external information in the evolutionary process of each algorithm. Then, we developed an improved version, combining knowledge from various validations and experimental analyzes. This version was then compared to a state-of-the-art hyper-heuristic for multi-objective optimization. The results were favorable to the proposed cooperative approach. Finally, we evaluate the proposed framework in a real-world problem. This problem optimizes the parameters of the construction of a wind turbine. The results achieved are comparable to those of the best MOEA evaluated for this problem. We conclude that the proposed information exchange method is an effective approach for the cooperation of several MOEAs for many-objective optimization. Besides, we conclude that it can achieve competitive results for a wide variety of problem instances.

Keywords: Many-objective Optimization, Hyper-heuristic, Continuous Optimization, Heuristic Selection, Evolutionary Algorithm, Real-world problem

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AOS	Adaptive Operator Selection
ASF	Achievement Scalarizing Function
AsyncHeDi	Asynchronous version of HeDi
BRACIS	Brazilian Conference on Intelligent Systems
CD	Crowding Distance
CEC	IEEE Congress on Evolutionary Computation
CF-HH	Choice Function based Hyper-heuristic
CMX	Center of Mass Crossover
dEA	Distributed Evolutionary Algorithm
DRA	Dynamic Resource Allocation
EA	Evolutionary Algorithm
FE	Fitness Evaluation
GECCO	Genetic and Evolutionary Computation Conference
HeDi	Heterogeneous Distributed cooperation of MOEAs
HH	Hyper-heuristic
HH-CO	Cooperative based Hyper-heuristics
HH-LA	Learning Automata based Hyper-heuristics
HHcMOEA	Hyper-heuristic guided cooperation of MOEAs
HHMOEA	Hyper-heuristic without the cooperation step
HMOPSO	Hyper-heuristic based Multi-objective Particle Swarm Optimization
HV	Hypervolume
IGD	Inverted Generational Distance
IQR	Interquartile Range
LLH	Low-level heuristic
MAB	Multi-armed Bandit
MaOP	Many-objective Optimization Problem
MOEA	Multi-objective Evolutionary Algorithm
MOHH	Multi-objective Hyper-heuristic
MOO	Multi-objective Optimization
NFL	No Free Lunch
SDE	Shift-based Density Estimation
SPX	Simplex Crossover
STM	Stable Matching Selection
SyncHeDi	Synchronous version of HeDi
WFG	Walking Fish Group

CONTENTS

1 INTRODUCTION

Multi-objective optimization (MOO) is a research field devoted to the investigation of techniques capable of optimizing problems with more than one objective function. These problems usually do not have a single best solution, as they frequently present conflicting objectives. Thus, the goal is to find a set of solutions representing the different trade-offs, considering all the objectives. Evolutionary Algorithms are often successfully applied to multi-objective optimization (called Multi-Objective Evolutionary Algorithms - MOEAs). However, when the number of objectives is higher than three, those problems are called many-objective problems (MaOP), and they require the use of specific strategies (Li et al., 2015a). This subcategory is characterized by the difficulties it presents, mainly due to the difficulty of distinguishing solutions. Many of them are not comparable. Also, the difficulty in visualizing the solution sets on more than three dimensions, among others (more details about many-objective optimization are presented at Section 2.1). Recently various MOEAs have been proposed for tackling this kind of problem (Li et al., 2018a). However, each one has different strengths and weaknesses, depending on the problem instance (Ishibuchi et al., 2017). Further, different benchmark problems have been proposed for many-objective optimization research in the last two decades, for example, DTLZ (Deb et al., 2005), WFG (Huband et al., 2005), and MaF (Cheng et al., 2017).

Furthermore, it is usually discussed that Pareto based evolutionary algorithms do not scale well when the number of objectives to be optimized increases. Therefore, different MOEAs were proposed, especially those based on decomposition, reference points, and indicators. However, the performance of a MOEA strongly depends on the properties of the problem instance being tackled, such number of objectives and Pareto front shape. Moreover, depending on the problem characteristics, Pareto based MOEAs can achieve results similar to or better than state-of-the-art many-objective evolutionary algorithms (Ishibuchi et al., 2017). Thus, choosing an appropriate algorithm to solve a problem becomes another difficult task to be solved.

In this scenario, the use of hyper-heuristic can contribute to solving many-objective optimization problems. Hyper-heuristics (HH) are high-level strategies aiming at selecting or generating low-level heuristics (LLH) (Burke et al., 2013, 2019; Drake et al., 2020). Those LLH may be, for example, reproduction or selection operators, local search algorithms, or MOEAs. In the online selection case, the HH task is choosing the heuristics to be applied for a given problem during the search. Moreover, the best performing LLH may be different for different search stages; therefore, the hyper-heuristic switches from LLHs as the evolution progresses (Guizzo et al., 2017). This type of procedure helps the integration of different strategies to solve a particular problem instance. It combines different approaches during the search to join strengths and to overcome the weaknesses they have separately. Besides, hyper-heuristics contribute to reducing the manual intervention of a specialist by providing efficient and reusable methodologies; applicable to a wide range of problems (Li et al., 2018b).

Hyper-heuristics assume that no single algorithm can solve every problem in a reasonable time, known as the No Free Lunch theorem. In other words, if one algorithm is the best for one problem, another algorithm is better for another. This behavior can be observed on many-objective optimization algorithms since their performance depends on the problem characteristics (Ishibuchi et al., 2017). Another example can be seen in the work of Li et al. (2018a), where 13 state of the art MOEAs are evaluated on seventeen test instances. It demonstrates that none of the algorithms being assessed outperform the others in all types of test instances. In fact, different algorithms presented advantages for various problems.

Similarly, in the Appendix C, we compared nine MOEAs. We demonstrated the presence of the no Free Lunch theorem for the evaluated benchmark suite and algorithms. Therefore, in this thesis, we propose and investigate the cooperation of multiple MOEAs and the use of hyper-heuristics to build generally applicable strategies to many-objective optimization.

Finally, the MOEAs proposed for many-objective optimization are usually not evaluated on real-world applications (Tanabe and Ishibuchi, 2020). As a consequence, there are only a few studies on many-objective real-world problems (Safi et al., 2018). A probable reason is the lack of well-established real-world problem formulations, mainly when dealing with continuous optimization (Tanabe and Ishibuchi, 2020). Recently, the 3rd Evolutionary Computation Competition presented a real-world many-objective problem (The Japanese Society of Evolutionary Computation, 2019). This problem is an instance of the wind-turbine design for wind power generation, a relevant renewable energy problem. It provides the problem model and description, as well as an evaluation module. Given the variable values, this module computes the objectives and constraints. In addition, it established a methodology for evaluating algorithms in this problem. We highlight that this problem, and most real-world problems, have constraints that must be satisfied. Commonly, research on many-objective evolutionary algorithms is not concerned about dealing with this topic (Fan et al., 2020). On the other hand, there are examples of how to extend dominance-based and decomposition-based MOEAs to deal with constraints (H. Jain and K. Deb, 2014; Fan et al., 2020). In this work, we evaluate our proposed approach applied to the Wind Turbine design problem.

1.1 GOAL

The main goal of this research is to improve the search quality of MOEAs for many-objective optimization concerning problems with a wide variety of characteristics. To this end, a collaboration model between algorithms is proposed. This collaboration is made by using hyper-heuristics to weight the participation of each MOEA during the search. Therefore we introduce a new hyper-heuristic framework for many-objective optimization. Its main characteristic is that the MOEAs exchange information during the execution. This manuscript describes the advantages of this new framework, as well as the research behind its conception.

1.2 METHODOLOGY

In this research, we use hyper-heuristics and exchange of information to guide the cooperation of multiple MOEAs to solve many-objective problems. We argue that the collaboration of different strategies to solve a problem may improve the search ability. Moreover, the hyper-heuristic weights the influence of each MOEA during the search process. The idea behind hyper-heuristics is to apply more often a MOEA that is showing the best performance and thus achieving better results than each MOEA alone. Furthermore, the aim is raising the generality level allowing the achievement of good results on a great set of instances than better results on a particular problem instance (Burke et al., 2013; Demeester et al., 2012). Another significant contribution of this work is evaluating the effect of using an information exchange on the hyper-heuristic procedure.

In detail, we started this research by proposing the execution of multiple MOEAs simultaneously, exchanging information during the search. The idea was to improve the generality by combining the strengths of different approaches to overcome its disabilities. The initial results of the cooperation were evaluated for the execution of two MOEAs simultaneously. The population was split in half for each MOEA. Moreover, the MOEAs were guided by weight vectors. Thus, leading each MOEA in different directions to cover the whole Pareto front. We evaluated it using synchronous and asynchronous communication. The initial results were promising, favorable to the synchronous communication, and instigated further research. However, the increase in the number of MOEAs revealed the challenges of using this approach. One of them is the population size, which was too small when the number of MOEAs was increased. Another difficulty was to include not decomposition-based strategies, as they do not lay on weight vectors to guide the search. The solution was to execute one MOEA at a time and weigh the participation of each MOEA. Those observations directed the research towards the use of hyper-heuristic, and a preliminary framework was proposed.

Then we present and evaluate the preliminary hyper-heuristic framework (HHcMOEA) to validate the ideas proposed after the initial experiments. This framework was used to assess and validate the importance of the information exchange step in the hyper-heuristic framework. We could also demonstrate that the best MOEA varies for different problem instances, advocating the use of hyper-heuristics. Therefore, using hyper-heuristics, we achieved the best, or equivalent to the best, average result in almost all problem instances. When the information exchange step is deactivated, the results are significantly degraded. This demonstrates the relevance of the proposed information exchange step.

Based on the preliminary analysis and further experimentation and research, we propose the Cooperative based Hyper-heuristic (HH-CO) for Many-objective Optimization. The HH-CO was first evaluated on the MaF benchmark set. This benchmark includes a diversity of characteristics, representing the challenges that real-world problems may pose to MOEAs. The HH-CO is compared to the MOEAs from its pool and a state-of-the-art hyper-heuristic, HH-LA (Learning Automata-Based Multiobjective Hyper-Heuristic). The results were favorable to HH-CO and instigated the study on a real-world application. Therefore, it was evaluated on the recently proposed wind-turbine design problem. Additionally, as it is a constrained problem, a constraint handling approach is incorporated into the algorithms. Thus, this study also contributes to understanding the problem characteristics and the behavior of state-of-the-art MOEAs in a real-world application. Finally, the HH-CO is favorable compared to a set of eight MOEAs[1].

1.3 CONTRIBUTIONS

1. This research proposes and demonstrates that the cooperation of multiple MOEAs can improve their searchability to achieve better results.

2. This research proposes and demonstrates that incorporating information exchange into a hyper-heuristic framework and the use of internal population for each low-level heuristic allows achieving better results in a wide range of problems.

3. This research demonstrates that the proposed approach is competitive to the state-of-the-art MOEAs used in its pool of low-level heuristics, competitive to a state-of-the-art hyper-heuristic framework for multi-objective optimization, and also competitive to state-of-the-art MOEAs that are not included in its pool.

4. This research demonstrates that the proposed approach is an effective strategy for real-world applications, considering that the best MOEA for the problem is not known in advance.

[1]The MOEA/DD had difficulties handling constraints and was not considered in the analysis

1.3.1 Published work

During this research, we proposed the study on the cooperation of MOEAs for many-objective optimization. An initial approach was proposed, using the island model for the collaboration between two MOEAs. The results were favorable and published in the annals of GECCO'17 (*Genetic and Evolutionary Computation Conference*) (Fritsche and Pozo, 2017a). Then, two communication strategies, one synchronous and one asynchronous, were evaluated. The conclusions obtained by this analysis were published in the annals of BRACIS'2017 (*Brazilian Conference on Intelligent Systems*) (Fritsche and Pozo, 2017b). Next, the online selection of algorithms was incorporated to guide the cooperation of multiple MOEAs. At the same time, they exchange information after being applied. The results obtained were published in the annals of BRACIS'2018 (Fritsche and Pozo, 2018). Based on the improvements achieved after extensive research and experimentation, a new approach was published in the annals of GECCO'19 (Fritsche and Pozo, 2019). Finally, this cooperative framework was evaluated on a real-world application, the wind turbine design problem. The results of this analysis were published in the annals of CEC'20 (*IEEE Congress on Evolutionary Computation*) (Fritsche and Pozo, 2020).

Moreover, the association with co-authors in related research led to the publication of two journal papers. First, a multi-objective and evolutionary hyper-heuristic applied to the integration and test order problem was published in the Applied Soft Computing (Guizzo et al., 2017). Also, an evaluation of selection methods for HMOPSO was published in the Journal of Heuristics (Castro et al., 2018).

We are currently working on a journal paper based on further experiments and analysis of the characteristics of the cooperative framework. The goal is to better understand and validate the results and conclusions presented in this manuscript and other published work. Those experiments include analysis of the choices made by the cooperative framework compared to a state-of-the-art hyper-heuristic. As well as their comparison in the wind turbine design problem.

1.4 TEXT ORGANIZATION

The remainder of this thesis is organized as follows:

- The Chapter 2 presents the background required to better understand the concepts applied in this thesis. For instance, we discuss many-objective optimization, multi-objective evolutionary algorithms, especially those used in this research. Also, we discuss other details, such as the generation of weight vectors, a method to handle constraints, and quality indicators used to assess the quality of the output from MOEAs. Regarding the methodology, for experimental analysis, we present background about benchmark problems, population size setting. Also, we discuss a real-world application, the wind turbine design problem. Finally, we discuss online selection hyper-heuristics.

- In Chapter 3, we present related works and discuss them in the context of this thesis. Initially, we present related research on many-objective optimization. Next, we discuss some research on the cooperation of different multi-objective optimization strategies, including hybrid approaches and distributed evolutionary algorithms. Then, we explore the related works on multi-objective hyper-heuristics, especially the heuristic selection for controlling multiple MOEAs. Followed by an overall discussion about this thesis in the context of the state-of-the-art literature.

- Chapter 4 presents the approaches we proposed for the cooperation of multiple MOEAs for many-objective optimization. The main characteristic is the proposed migration procedure that allows exchanging information among MOEAs. Also, we present the MOEAs used in this research. Then, we present the first proposed approach for the cooperation of MOEAs running simultaneously. Next, we present our second proposed approach, where the proposed migration method is included into a hyper-heuristic. Finally, we present our final proposed approach based on the knowledge achieved during this research.

- The Chapter 5 presents the main experimental analysis made during this research and the main achieved observations. We present the experimental setup and the four main experiments. First, we present the validation for the distributed cooperation of two MOEAs. Next, we explored a preliminary analysis with hyper-heuristics with and without using the proposed migration procedure. Finally, we present the analysis for the current version of the cooperative hyper-heuristic. We compare it to a state-of-the-art hyper-heuristic and the winners from the CEC'18 competition. Moreover, we examined the preliminary and the current proposed version. In the end, we evaluate the cooperative hyper-heuristic on a real-world application.

- In Chapter 6, we briefly review the main aspects of this work and present the main observations and conclusions. Also, we present guidelines for future works.

- In the appendices, we present complementary experiments with the distributed cooperation of MOEAs, followed by proof of concepts to initially explore the use of hyper-heuristics, and experiments demonstrating the No Free Lunch for many-objective optimization.

2 BACKGROUND

In this chapter, we introduce definitions and discuss some difficulties in solving many-objective optimization problems. We describe a classification for multi and many-objective evolutionary algorithms. Next, we present the strategies used in this study to deal with constraints. Moreover, we introduce some real-world many-objective applications and their main characteristics. Finally, it is discussed online selection hyper-heuristics and their application to multi and many-objective optimization.

2.1 MANY-OBJECTIVE OPTIMIZATION

A multi-objective problem, represented by equation (2.1), searches for a set of values for n decision variables ($\mathbf{x}$), every one bounded by a lower (L) and upper (U) value, that minimizes M objective functions simultaneously. Further, the solutions must satisfy K equality and J inequality constraints, such that $K, J \geq 0$. Due to the duality principle, maximization problems can become minimization (Sun et al., 2019b). Therefore, in this manuscript, all examples and definitions consider the minimization case.

$$\begin{array}{ll} \text{Minimize} & (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})), \\ \text{subject to} & g_j(\mathbf{x}) \geq 0, \quad j = 1, 2, \ldots, J, \\ & h_k(\mathbf{x}) = 0, \quad k = 1, 2, \ldots, K, \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, \ldots, n. \end{array} \tag{2.1}$$

Usually, there is conflict among the objectives, what difficulties the comparison of solutions. A pair of objectives is in conflict when the improvement of one results in a deterioration of the other. One way of comparing solutions is using Pareto dominance, that allows classifying the relationship of two different solutions into: (i) *a dominates b*, if a is better or equal than b in all objectives, and strictly better in at least one; then, (ii) this solution b is said *dominated* by a; otherwise, (iii) a and b are *non-dominated*. An example of multi-objective problems is represented by Figure 2.1. In the example, a dominates b, since a is better (has smaller objective function value) in both f_1 and f_2 objectives. However, a and c are non-dominated compared to each other, since c is better than a for the objective f_1, while a is better than c for objective f_2.

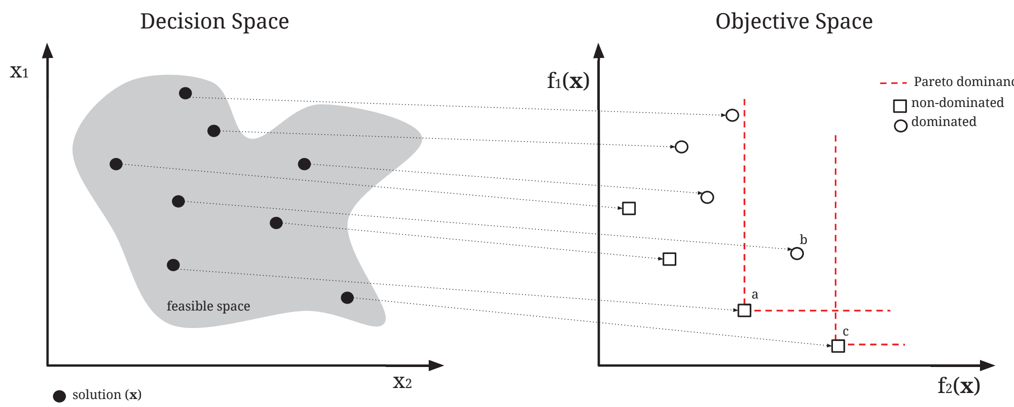


Figure 2.1: Example of multi-objective problem

Formally, for multi-objective optimization, the following definitions apply (Xue et al., 2019):

Definition 1. (Pareto Dominance) Considering a minimization problem, and the decision vectors (solutions) a and b. a is said to *dominate* b, same as b is dominated by a (denoted as $a \prec b$), if and only if

$$\forall\, i \in (1, 2, \ldots, m) : f_i(a) \leq f_i(b) \wedge \exists\, i \in (1, 2, \ldots, m) : f_i(a) < f_i(b) \tag{2.2}$$

In other words, *a dominates b* if its objective values are better or equal than all the respective objective values of b, and strictly better in at least one.

Definition 2. (Pareto Optimality) A decision vector a is said Pareto-optimal if and only if there is no decision vector b that dominates it.

Definition 3. (Pareto Set) Considering a given multi-objective problem, the Pareto Set is the set of all Pareto-optimal solutions in the decision space (decision vectors).

Definition 4. (Pareto Front) the Pareto Set image in the objective space of a given multi-objective problem is called the Pareto Front (PF).

Due to multi-objective characteristics, there is a set of equally optimal solutions for the problem. However, the decision-maker will effectively choose only one of these solutions, based on preferences. The preferences can be incorporated *a priori*; during the search (using, for example, interactive strategies), or *a posteriori* (Miettinen, 1998). In this work, we focus on setting the decision-maker preferences *a posteriori*. That is, the optimization will return a set of solutions from which the decision-maker will choose. This optimization aims to find an approximation of the entire Pareto front. Therefore, the goal is to approximate the optimal set of feasible (decision space) non-dominated (objective space) solutions. This set of solutions must hold two properties, represented by Figure 2.2. The first is proximity, which aims at finding an approximation set of non-dominated solutions as close as possible to the optimal Pareto front set. The other is completeness; the solutions must represent different trade-offs among the objectives. Therefore, the algorithms must find an approximation set that exhibits proximity and completeness to the Pareto front (Coello et al., 2007).

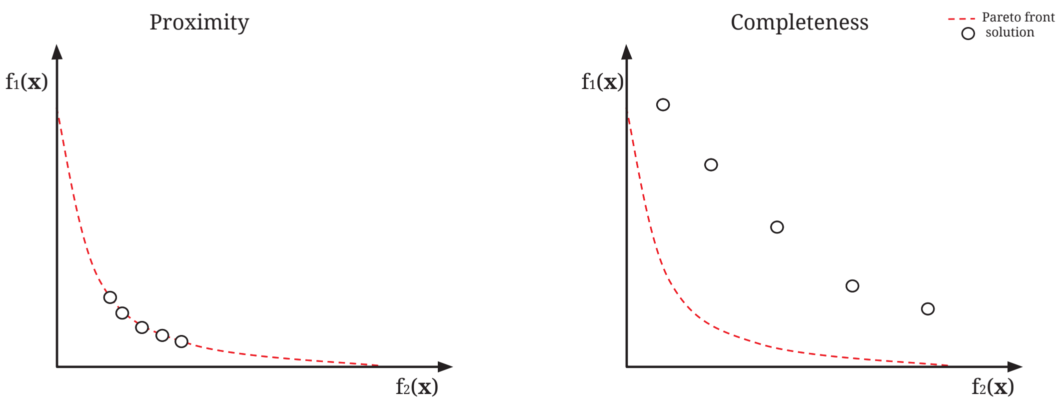
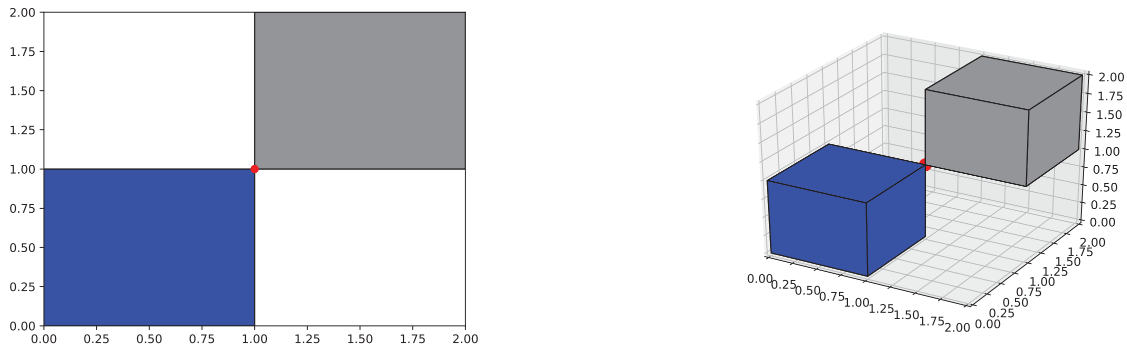


Figure 2.2: Example of proximity versus completeness

A many-objective problem is a problem where the number of objectives is higher than three. This subcategory has received the attention of the researchers due to the difficulties it presents (Ishibuchi et al., 2008; Figueiredo, 2013; Deb and Jain, 2014; Li et al., 2015a; Sun et al., 2019a; Wan et al., 2019). First, there is a lack of selection pressure since, in a high-dimensional space, most solutions become non-dominated to each other. Thus, the algorithm finds difficulties distinguishing which solution is better than others, thus degrading the search ability. For instance, parent selection may become random since most solutions are not comparable, losing convergence pressure. Also, environmental selection may be affected, as the new solutions generated also tend to be non-dominated.

An illustration of the increasing of non-dominated proportion as higher the number of objectives is demonstrated by Figure 2.3. In this figure, we placed a red point in the center of the objective space. The blue area represents the space that dominates it. On the other hand, the gray area is dominated by the point. While the white (or empty) space represents the amount of non-dominated. For two objectives (Figure 2.3(a)), 25% of the space dominates, other 25% is dominated, while 50% is non-dominated compared to the point. Next, we increased the number of objectives to three (Figure 2.3(b)). In this case, only 12.5% of space dominates the point, while the other 12.5% is dominated by it, while 75% of the space is non-dominated. This example demonstrates how the amount of non-dominated solutions may rapidly increase as higher the number of objectives. Nonetheless, other characteristics challenge the algorithms in improving non-dominated solutions such as multi-modality and discontinuities.



(a) Two objectives

(b) Three objectives

Figure 2.3: Example of higher non-dominated space increasing the number of objectives

Also, the computational cost for both fitness evaluation and the algorithm operations increases considerably. Besides, the crossover operation becomes inefficient, as parents tend to be distant from one another. Moreover, it is hard to maintain diversity since the required number of solutions representing the Pareto front grows exponentially as the number of objectives increases. Finally, when the dimensionality is higher than three, the visual representation is problematic, which challenges the decision-maker at choosing the final solution. For this latter issue, specific research has been applied — for example, the research from Xiong et al. (2019).

2.1.1 Multi and Many-objective Evolutionary Algorithms

Multi-objective problems may present Pareto front with different shapes due to non-linearities, multi-modalities, or others. Concavities in the objective front are quite common on many real problems, leading to discontinuities in the Pareto front. Therefore, derivative-based methods may have difficulties in optimizing them. Evolutionary Algorithms (EAs) are particularly suitable for multi-objective optimization with conflicting objectives. This is due to their population-based meta-heuristic, gradient-free, black-box characteristics (Wan et al., 2019; Sun et al., 2019a; Safi et al., 2018). EAs can evaluate a set of solutions simultaneously and output a set of solutions with different trade-offs between objectives in a single run. Further, EAs have a relatively low computational cost to get a good approximation set and solve several optimization problems.

When applied to multi-objective problems, the EAs are called MOEAs. Usually, a general MOEA framework follows the steps presented at Algorithm 1 (Xue et al., 2019). Initially, a population of size p is generated randomly. Then, until the stop criterion is met, the following steps repeat. First, the mating selection will choose the best individuals to compose the next generation parent set. The idea is to push for quality improvement, as the combination of good parents is more likely to generate good offspring. Then, the reproduction takes place, creating new solutions by applying transformations considering the parents, according to the rules of each algorithm (e.g., crossover and mutation). The reproduction produces offspring of size o. Finally, the environmental selection filters the solutions $p + o$ to get the population for the next generation. The individuals are filtered based on the rules of each algorithm, frequently using Pareto dominance.

Algorithm 1: A general MOEA framework

```
population initialization;
while the stop criteria is not met do
    mating selection;
    reproduction; // crossover + mutation
    environmental selection;
end
```

Usually, Pareto-based MOEAs are successful for problems with two and three objectives. However, by increasing the number of objectives, new difficulties arise, as commented before. Due to these difficulties, different types of MOEAs were proposed (Ishibuchi et al., 2008). They can be classified into different categories (Li et al., 2015a):

- Pareto dominance: These algorithms use Pareto dominance as the first criterion for comparing solutions, such as NSGA-II and SPEA2. Pareto dominance based MOEAs have been extensively reported as not having good results for many-objective optimization. However, depending on the problem characteristics, they may achieve results as good or better than state-of-the-art many-objective evolutionary algorithms (Ishibuchi et al., 2020).

- Aggregation-based: Such algorithms, e.g., MOEA/D, do not suffer from the lack of selection pressure as they do not use the concept of Pareto dominance. However, they still have difficulties due to the curse of dimensionality. Also, the adequate distribution of weight vectors is not a trivial task.

- Indicator-based: such as SMS-EMOA, HypE, MOMBI-II, and IBEA. These MOEAs aim at maximizing the value of a given quality indicator. However, the high computational cost associated with the calculation of the indicators may become an issue.

- Preference-based: In this category, the decision-maker may include its preferences during (interactive) or before the search. Therefore, reducing the complexity of the problem. For example, the research of Wang et al. (2019). However, it may be difficult for the decision-maker to properly configure this kind of approach.

- Dimensionality reduction: in this category, the objective is to reduce the complexity of the problem, for example, by removing redundant objectives. This strategy is limited to problems where the optimization of the simplified problem represents the original characteristics.

- Relaxed dominance: in this category, the objective is to change the dominance criterion to increase the selection pressure. The difficulty with these methods is in the configuration of the parameters that control relaxation. Recently, Liu et al. (2020) propose an angle dominance criterion exempt from the parameter tuning.

- Modified diversity mechanism: for example, the Shift-based Density Estimation (SDE), in this case, the convergence information is considered when evaluating diversity. The goal is to avoid the "diversity only" effect that may occur when Pareto dominance fails to distinguish solutions.

- Hybrid strategies: these strategies combine two or more approaches to overcome their individual difficulties. Some examples are NSGA-III (Deb and Jain, 2014), which combines Pareto dominance and aggregation; and Two_Arch2 (Wang et al., 2015a), based on an indicator, Pareto dominance, and diversity.

2.1.2 Multi-objective optimization research over the past decades

Multi-objective optimization has been an active topic of research in the past decades. This section presents some highlights of the proposed multi-objective evolutionary algorithms and their characteristics and benchmark problem sets. In Figures 2.4, 2.5 and 2.6 a history is presented containing some of the main multi-objective meta-heuristics and benchmark problem sets.

The reference problems used around the 1990s did not present any difficulty in convergence. Even randomly generated solutions were close to the Pareto front (Ishibuchi et al., 2017). Therefore, algorithms such as NSGA (Srinivas and Deb, 1994), which do not have strong convergence pressure, were proposed at that time. With the emergence of new benchmark problems, such as ZDT (Zitzler et al., 2000), the initial random solutions are far from the Pareto front but have good diversity. Then elitist algorithms appeared, such as SPEA2 (Zitzler et al., 2001) and NSGA-II (Deb et al., 2002), improving the convergence of their precursors.

Some years later, the first sets of problems for many-objective optimization emerged: the DTLZ (Deb et al., 2005) and the WFG (Huband et al., 2005) benchmarks. These problems have been extensively used to demonstrate that Pareto dominance-based MOEAs do not perform well with a high number of objectives. Therefore, algorithms based on indicators were proposed to overcome this difficulty, such as IBEA, HypE, SMS-EMOA, and MOMBI. The main disadvantage is the computational cost.

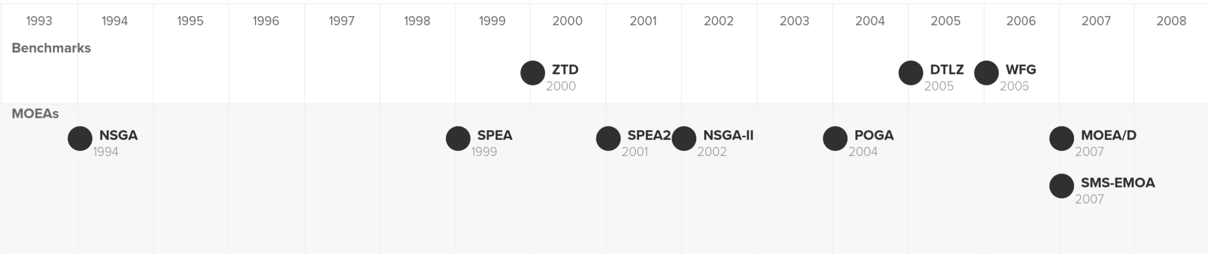


Figure 2.4: History of some of the main MOEAs and benchmarks over the years (Part I: from 1994 to 2007).

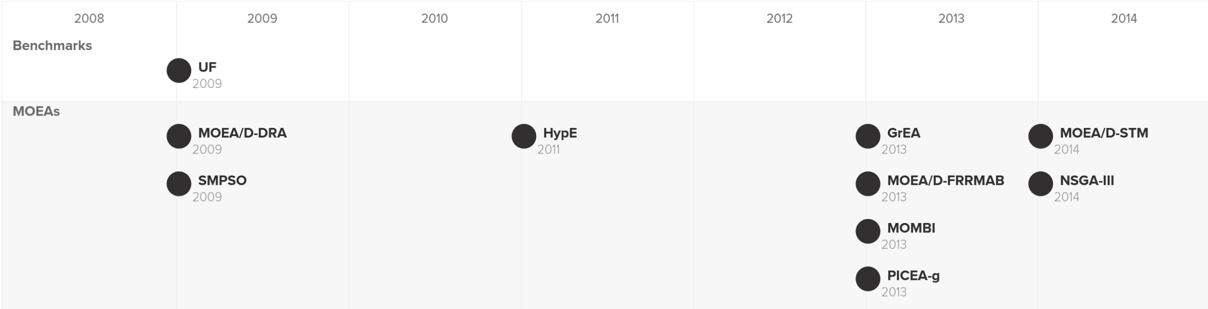


Figure 2.5: History of some of the main MOEAs and benchmarks over the years (Part II: from 2008 to 2014).

In 2007, it was proposed the multi-objective evolutionary algorithm based on decomposition (MOEA/D). Initially, it was not intended for many-objective optimization. Still, MOEA/D good performance in this type of problem triggered the creation of several variations (MOEA/D-FRRMAB, MOEA/D-DU, MOEA/D-STM, EFR-RR, MOEA/DD). However, their performance is degraded when the Pareto front does not match the shape of the distribution of the weight vectors used (Ishibuchi et al., 2017). Furthermore, the properties of some benchmark problems (DTLZ and WFG) make these not hard for this type of algorithm (Ishibuchi et al., 2017). For example, in the WFG and DTLZ, some variables only affect convergence (called distance variables). Modifying the value of the distance variables will not affect diversity. Also, some variables do not affect convergence, called position variables. For some problem instances, if a solution is Pareto optimal, all solutions generated by changing only the position variables will also be optimal. Therefore, it is easy to improve the diversity of solutions without deteriorating convergence. Likewise, improving convergence is also not difficult, for example, using a high penalty value in the PBI scalar function. An algorithm that exploits this characteristic is MOEA/D-DI (He and Yen, 2014). First, a weight vector is used to optimize each objective separately; to guide the entire population towards the Pareto front (convergence). After this convergence phase, subpopulations are initialized around each solution found; those solutions are then evolved to find a broad distribution (diversity).

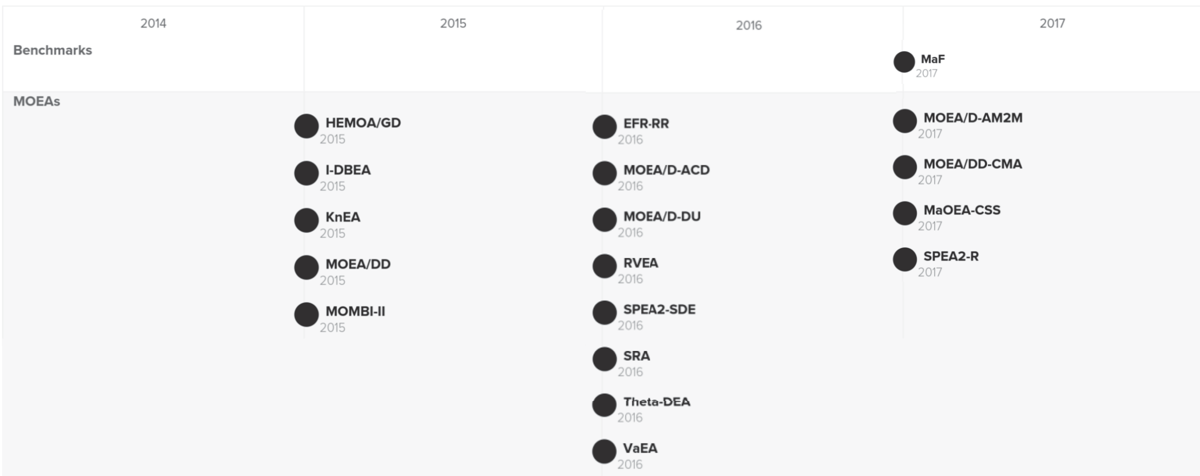


Figure 2.6: History of some of the main MOEAs and benchmarks over the years (Part III: from 2015 to 2017).

More recently, hybrid approaches between Pareto dominance and decomposition have also demonstrated good quality with many objectives, e.g., MOEA/DD and NSGA-III. These algorithms have shown excellent performance in the DTLZ, and WFG sets, in instances from 2 to 15 objectives. Furthermore, the MaF benchmark (Cheng et al., 2017) was recently proposed, comprising a set of 15 problem instances, representing different challenges from many-objective optimization. The MaF benchmark is presented in detail in Section 2.1.7.

2.1.3 MOEAs used in this research

In this research, we use a set of ten MOEAs, including different MOEA categories:

- SPEA2 (Zitzler et al., 2001): the SPEA algorithm (predecessor of SPEA2) was one of the first to introduce elitism in the search to better keep valuable solutions. In SPEA, an external repository is used to keep the non-dominated solutions. If the file size limit is exceeded, it is truncated using a diversity criterion. The objective of SPEA2 was to overcome some difficulties encountered by the authors in their initial version. The main differences from SPEA2 to SPEA are: a new strategy for assigning quality to offspring, considering how many solutions it dominates, and how many it is dominated by; a technique based on the nearest neighbor is used to evaluate the density of solutions (diversity); and, it adds a new truncation method, which guarantees the preservation of extreme solutions. Also, in SPEA2, only the solutions from the external file are used in parent selection.

- NSGA-II (Deb et al., 2002): this is probably one of the most well-known MOEA[1]. Based on a genetic algorithm, this MOEA uses an elitist strategy to update the population. First, genetic operators are applied to the parent population, generating offspring. Then, the population and offspring are combined into a single set, sorted according to the Pareto dominance criterion. After sorting, the solutions are included in the new population following their Pareto front rank. When a given front cannot be fully included (given the population size limit), the diversity criterion is applied to distinguish solutions. In NSGA-II, the diversity criterion used is the Crowding Distance (CD). It measures the maximum hypercube size that encapsulates a solution without including any other. Solutions with higher CD values are preferable, while solutions with lower CD values are removed.

- MOEA/D (Zhang and Li, 2007): This algorithm incorporates the objective space decomposition strategy into a multi-objective evolutionary algorithm. In the decomposition, the objective space is divided into several single-objective subproblems, optimized simultaneously. This algorithm maintains a population in which each solution is associated with a weight vector. Also, each weight vector is linked to a neighborhood of size defined by a parameter. In MOEA/D, the parent selection is made randomly in the neighborhood. Then, the parents are combined using genetic operators to generate offspring. After the fitness evaluation, the solution that minimizes the scalar function (for example, Tchebycheff) is associated with each weight vector. This algorithm has been extensively studied in the literature. Several variations are proposed and applied in the optimization with many objectives, obtaining excellent results on benchmark problems (Ishibuchi et al., 2017).

- HypE (Bader and Zitzler, 2011): it is similar to the NSGA-II using Pareto dominance as the first criterion to compare solutions. However, the second criterion is based on the contribution of the solutions to the hypervolume. Moreover, an approximation of the hypervolume is used to reduce the computational cost. The authors advocate that the exact hypervolume value is not significant, but the ranking of solutions is.

[1]According to IEEE Xplore, the NSGA-II paper was cited over 19,000 times until October 2020.

- MOEA/D-STM (Li et al., 2014b): It is an evolutionary algorithm based on decomposition. The main feature of the algorithm is the Stable Matching Selection (STM) to match a weight vector with a solution. The STM is the strategy of selection for replacement, to compose the individuals of the next generation. First, a weight vector selects a solution that minimizes its aggregation function and has not yet been paired and proposes to pair. The solution chooses the closest one between its current pair (if any) and the proponent. The MOEA/D-STM also uses Dynamic Resource Allocation. At every iteration, the algorithm selects the vectors set to be updated using a 10-tournament process based on a utility function. The algorithm updates the function value after a given number of iterations. For the reproduction phase, the algorithm uses Differential Evolution (DE) and polynomial mutation. It uses random parent selection, with a probability of selecting from the neighborhood or the whole population.

- NSGA-III (Deb and Jain, 2014): it is similar to the NSGA-II in the use of Pareto dominance. However, the most significant difference is in the diversity criterion. The NSGA-II applies the CD, while NSGA-III uses a set of reference points after applying the Pareto dominance criterion. Initially, the solutions are normalized (with bounds adaptively set throughout the search). Then, each solution is associated with the nearest reference point. Then, it counts how many solutions are associated with each point (niche count). Finally, a set of rules decides which solutions will be included in the next population.

- SPEA2SDE (Li et al., 2014c): it modifies the diversity maintenance mechanism of SPEA2, using a shift-based density estimation (SDE) strategy. The SDE considers both the distribution and convergence of the solutions. The SDE goal is to place solutions with poor convergence into crowded regions. Therefore, they will be eliminated from the population. The SDE strategy can be incorporated in different MOEAs. However, the SPEA2 with SDE was the most competitive from the options evaluated.

- MOEA/DD (Li et al., 2015c): it is based on both Pareto-dominance and decomposition. It is a steady-state approach; it means that the population is updated every time a new solution is generated. In this update method, the MOEA/DD uses a set of rules. Initially, it computes the Pareto dominance rank (non-dominated sorting). Then, it associates the new solutions with the closest subproblem, using angular distance. The first criterion to remove a solution is Pareto dominance. The second criterion is the diversity of solutions (niche count). The third criterion is to minimize the scalarizing function (PBI). However, if the solution to be removed is the only representative of a given subproblem, the algorithm prefers to maintain a dominated solution, to preserve diversity.

- MOMBI2 (Gómez and Coello, 2015): In this algorithm, the solutions are evaluated by an R2-based ranking. The R2 has a lower computational cost than the hypervolume used in other indicator based MOEAs. It uses a set of weight vectors and the Achievement Scalarizing Function (ASF) to evaluate the solutions. The ASF value of all solutions is calculated for each weight vector. Then, the solutions are ranked by the ASF value for each weight vector. Finally, the quality of a solution is the best rank it obtained for the different weight vectors.

- ThetaDEA (Yuan et al., 2016): it is an extension of the NSGA-III, aiming at improving its convergence in problems with many objectives. For this, the Pareto dominance was replaced by a PBI-based dominance, called Theta dominance. Another difference from the NSGA-III is how ThetaDEA estimates the nadir point, necessary for normalization. First, in the Theta dominance, each solution is associated with the closest subproblem (weight vector), calculated by the perpendicular distance. Then, the quality value of the solution is obtained by calculating the PBI, using the reference vector to which the solution was associated. Finally, the fast nondominated sorting approach is applied to obtain different levels of Theta dominance. A solution is said to dominate another if both are associated with the same subproblem, and the PBI value of the solution is better than the PBI value of the other.

These ten algorithms are representative of different classes of MOEAs. Furthermore, the research from Li et al. (2018a) demonstrates that their performance, compared to one another, depends on the problem at hand. That research evaluates different MOEAs in several many-objective problems. As expected, no algorithm obtained the best results for all problems. Besides, MOEAs considered inefficient for many-objective problems (such as NSGA-II) achieved better results than state-of-art MOEAs, in some problems. Those observations are compatible with the No Free Lunch (NFL) theorem (Wolpert and Macready, 1997). One way to alleviate the NFL is to combine a set of criteria into the same algorithm. Another way to mitigate this issue is through the use of hyper-heuristics. For more information about many-objective optimization, see Li et al. (2015a) and Ishibuchi et al. (2008).

2.1.4 Weight vector generation

Several strategies for optimizing problems with many objectives use a set of weight vectors (or reference points) to guide the search. However, the distribution of the weight vectors in the objective space directly affects the performance of MOEAs (Ishibuchi et al., 2017). A widely used strategy is the one proposed by Das and Dennis (1998). It is possible to calculate the number of points using the Equation 2.3. m is the number of objectives, and p is a parameter that defines the number of divisions per objective, with a uniform distance of $\delta = 1/p$ (as shown in Figure 2.7). To generate points inside, not only in the borders, the value of p should be greater than the number of objectives m (Li et al., 2015c). However, in high dimensions, the number of points required by this strategy becomes very large. For example, if $m = 8$ and $p = 8$, then $H = 6,435$. A two-layer weight vector generation strategy was proposed to deal with this difficulty for problems with 8 or more objectives (Deb and Jain, 2014). In this strategy, first, an external layer is created; then, a secondary layer is generated and shrunk to cover the internal region of the objective space (Li et al., 2015c; Deb and Jain, 2014). An example is illustrated in Figure 2.8.

$$H = \binom{m + p - 1}{p} \tag{2.3}$$

2.1.5 Constraint handling

Only a few studies handle constraints in many-objective optimization (H. Jain and K. Deb, 2014; Fan et al., 2020). In this research, we used the following strategy: for dominance-based algorithms (including Pareto-dominance such as NSGA-II, and other dominance relations, such as ThetaDEA and MOMBI2), the method computes the objective values and the violation of each constraint — i.e., how far the constraint value was from being feasible. After, it computes the overall constraint violation of the solution, i.e., the sum of the violations for every constraint.

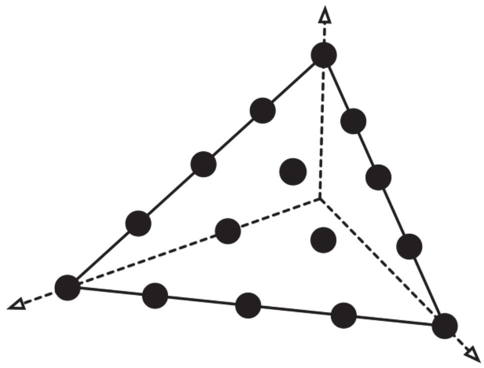


Figure 2.7: Example of weights generated by the Das and Dennis (1998) strategy, using the parameters: $p = 4$ and $m = 3$ ($H = 15$). Figure adapted from Li et al. (2015c)

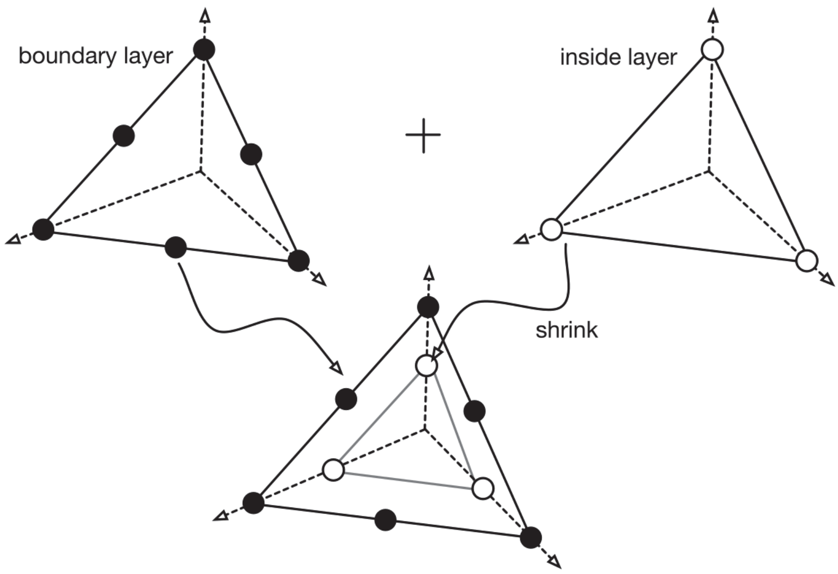


Figure 2.8: Example of weights generated by the two-layer strategy, using the parameters: $p_{external} = 2$, $p_{internal} = 1$ and $m = 3$ ($H = 9$). Figure adapted from Li et al. (2015c)

Finally, every time that the algorithm needs to compare two solutions ($\mathbf{x}_1, \mathbf{x}_2$), this method defines when $\mathbf{x}_1$ dominates $\mathbf{x}_2$ by (H. Jain and K. Deb, 2014; Fan et al., 2020):

1. $\mathbf{x}_1$ is feasible and $\mathbf{x}_2$ is infeasible,
2. $\mathbf{x}_1$ and $\mathbf{x}_2$ are infeasible and $\mathbf{x}_1$ has smaller constraint violation value, or
3. $\mathbf{x}_1$ and $\mathbf{x}_2$ are feasible and $\mathbf{x}_1$ dominates $\mathbf{x}_2$ (using some dominance relation).

Similarly, this method can be extended for decomposition-based MOEAs. It is done by replacing the dominance relation with the aggregation function in the third condition. We included this constraint handling method into eight of the algorithms used in this research. The MOEA/DD was the only one not compatible with this approach. It may favor dominated solutions, infeasible in this case, in favor of diversity. Moreover, the study of constraint handling approaches specifically for MOEA/DD is not the scope of this research.

2.1.6 Quality indicators

Several quality indicators have been proposed to assess the quality of approximation fronts generated by MOEAs and compare them. A quality indicator may evaluate convergence or diversity, or both simultaneously. In this research, we use the following quality indicators, all measuring both convergence and diversity simultaneously.

- **R2:** measures the average of the best aggregation value for each weight vector given an aggregation function. Compared to the hypervolume, the R2 presents less computational cost (being better suited to be used during the search); and it is assumed to favor uniformly distributed fronts (Brockhoff et al., 2012). Usually, the Tchebycheff aggregation function is used. However, others may be used, like ASF (Achievement Scalarizing Function), for example, (Gómez and Coello, 2015).

- **Inverted Generational Distance (IGD):** It measures both convergence and diversity on a single scalar, as smaller the better (Cheng et al., 2018; Li et al., 2015c). This indicator computes the distance from the true Pareto front (P^*) to the approximation front (P) generated by the algorithm. In detail (Figure 2.9(a)), it computes the average distance of the points in P^* to their closest points in P, defined as:

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|}, \tag{2.4}$$

  where $d(v, P)$ is the minimum distance from the point v to any point in P. Its main drawback is the requirement of a discrete representation of the true Pareto front, which is not available in real-world problems.

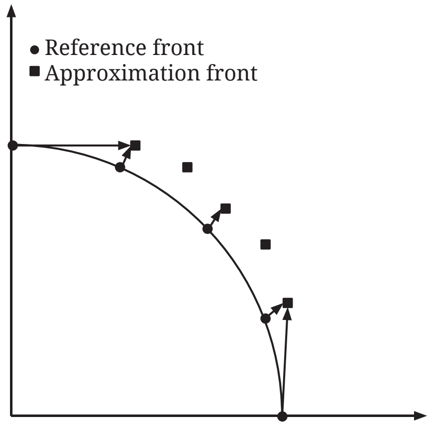
- **Hypervolume (HV):** computes both convergence and diversity in a single scalar (Cheng et al., 2018; Li et al., 2015c; Ishibuchi et al., 2017). However, it does not require the knowledge of the true Pareto front. Instead, it measures the high dimension volume of the space dominated by the solutions set, bounded by a reference nadir point (Figure 2.9(b)). As higher this volume, the better the approximation front is. Moreover, the hypervolume is a strictly monotonic metric. That means that if an approximation front dominates another, its hypervolume will be better. However, it has an expensive computational cost, and an approximated hypervolume is typically used for many-objective problems.

2.1.7 Benchmark problems

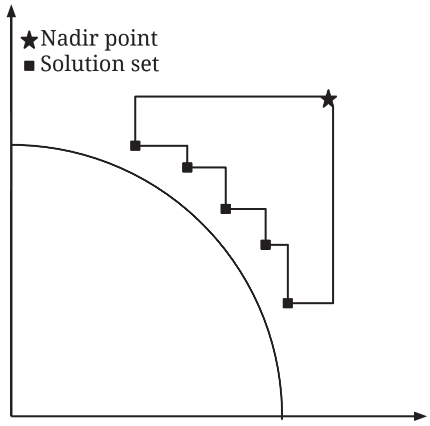
This section presents different families of benchmark problems and the main characteristics of each one.

2.1.7.1 DTLZ

Proposed by Deb et al. (2005), this set of seven problems has the following characteristics. First, it is possible to configure the number of decision variables and also the number of objectives. Another aspect is that all objectives (in the same problem) have the same scale. The properties of each problem are shown in Table 2.1 (Li et al., 2016).



(a) IGD



(b) Hypervolume

Figure 2.9: Example of IGD and Hypervolume computation

Table 2.1: DTLZ benchmark problem family properties

Problem	Properties
DTLZ1	Linear, Multimodal
DTLZ2	Concave
DTLZ3	Concave, Multimodal
DTLZ4	Concave, Biased
DTLZ5	Concave, Degenerate
DTLZ6	Concave, Degenerate, Biased
DTLZ7	Mixed, Discontinuous, Multimodal

2.1.7.2 WFG

This set, proposed by the *Walking Fish Group* - WFG, has nine instances (Huband et al., 2005). In this set, it is also possible to configure the number of objectives and decision variables. The scale between the objectives varies, which makes it difficult for MOEAs that do not incorporate normalization. The properties of these problems are shown in Table 2.2 (Li et al., 2016).

Table 2.2: WFG benchmark problem family properties

Problem	Properties
WFG1	Mixed, Biased
WFG2	Convex, Discontinuous, Nonseparable
WFG3	Linear, Degenerate, Nonseparable
WFG4	Concave, Multimodal
WFG5	Concave, Deceptive
WFG6	Concave, Nonseparable
WFG7	Concave, Biased
WFG8	Concave, Nonseparable, Biased
WFG9	Concave, Nonseparable, Deceptive, Biased

2.1.7.3 MaF benchmark problem set

Several benchmark problems have been proposed to support the multi-objective research, such as the DTLZ (Deb et al., 2005) and WFG (Huband et al., 2005). However, some characteristics of those benchmark problems make it easy for MOEAs to exploit those characteristics and therefore do not generalize for real-world problems. To avoid this issue, one needs to use a benchmark set with a diversity of properties, representing the different challenges that real-world many-objective optimization presents. The properties of the MaF benchmark problems are shown in Table 2.3 (Cheng et al., 2017).

Table 2.3: Properties of the MaF reference family problems

Problem	Properties
MaF01	Linear, Inverted
MaF02	Concave
MaF03	Convex, Multimodal
MaF04	Concave, Multimodal, Inverted
MaF05	Convex, Biased
MaF06	Concave, Degenerate
MaF07	Mixed, Disconnected, Multimodal
MaF08	Linear, Degenerate
MaF09	Linear, Degenerate
MaF10	Mixed, Biased
MaF11	Convex, Disconnected, Nonseparable
MaF12	Concave, Nonseparable, Biased, Deceptive
MaF13	Concave, Unimodal, Nonseparable, Degenerate
MaF14	Linear, Partially separable, Large scale
MaF15	Convex, Partially separable, Large scale, Inverted

The MaF is the benchmark problem set and methodology of the Competition on Many-Objective Optimization at 2017 and 2018 IEEE Congress on Evolutionary Computation (Cheng et al., 2017, 2018). The benchmark includes fifteen problems with diverse characteristics, such as linear, concave, convex, mixed shapes, multimodal, biased, degenerate, disconnected, and badly-scaled fitness landscapes. Every problem has an instance for five, ten, and fifteen objectives (a total of 45 problem instances).

This benchmark defines a methodology for evaluating and comparing MOEAs. Every MOEA is limited by the same number of fitness evaluations: $fe_{max} = \max(100000, 10000 \times D)$, being D the number of decision variables. The number of variables is defined for each problem (Cheng et al., 2018). Additionally, the number of independent runs is 20 (Cheng et al., 2018; Deb and Jain, 2014; Li et al., 2015c). The methodology of the competition evaluates the MOEAs using both IGD and Hypervolume. In this benchmark, the objectives are first normalized as seen in Equation 2.5:

$$f_i^j = \frac{f_i^j}{1.1 \times y_i^{nadir}}, \tag{2.5}$$

where, f_i^j is the i-th objective value of the j-th solution, and y_i^{nadir} is the i-th objective value of the nadir point (the worst value for each objective in the true Pareto front). Then, the reference point used to compute the hypervolume is the point $(1, \ldots, 1)$.

2.1.8 Setting the population size

Depending on the MOEAs properties, it is not possible to set an arbitrary number as population size. For example, MOEA/D (Zhang and Li, 2007) requires the population size to be equal to the number of weight vectors. However, the number of weight vectors cannot be arbitrarily set when using the approach of Das and Dennis for weight vector generation (Li et al., 2015c; Deb and Jain, 2014).

For many-objective optimization, two representative examples for setting this value are NSGA-III from Deb and Jain (2014) and MOEA/DD from Li et al. (2015c). The population size was set depending on the number of objectives. When this number is odd, but the MOEA being applied requires the population size to be even, this number can be incremented by one. This population size setting is presented in Table 2.4. The p is a parameter for the weight vector generation (see Deb and Jain (2014) and Li et al. (2015c) for more information about generating weight vectors). Besides, the population size of NSGA-III represents in this table the MOEAs that requires to be even, and MOEA/DD exemplifies population sizes when this is not required.

m	weight vectors	NSGA-III	MOEA/DD
3	91 ($p = 12$)	92	91
5	210 ($p = 6$)	210	210
8	156 ($p_1 = 3, p_2 = 2$)	156	156
10	275 ($p_1 = 3, p_2 = 2$)	276	275
15	135 ($p_1 = 2, p_2 = 1$)	136	136

Table 2.4: Population size for different objective number configurations

2.1.9 Real-world applications

There are in the literature examples of real-world many-objective applications. These examples include the engineering design (Fleming et al., 2005), a generic formulation aiming at minimizing a set of constraints. Other examples are the air traffic control problem, this problem has hard (must be satisfied) and soft (must be minimized) constraints (Herrero et al., 2009); scheduling problems, such as nurse rostering (Sülflow et al., 2007), with more than 20 objectives; the radar waveform design (Hughes, 2007), with nine objectives and an integer decision space; hybrid car controller (Rodemann et al., 2015), with seven objectives; space trajectory design (Jaimes et al., 2013) that has up to six objectives; vehicle routing ($M = 6$) (Safi et al., 2018). Other many-objective examples are the dispatch of produced electrical power and some problems from Search-Based Software Engineering (SBSE) (Safi et al., 2018). A relevant example is the water resource planning (Musselman and Talavage, 1980), with five objectives and seven constraints. This problem uses recorded precipitation data for planning storm-drainage systems for West Lafayette city. Some works used this problem as validation, together to benchmark problems (H. Jain and K. Deb, 2014).

Most of these examples are discrete problems. The continuous problems usually fall into the engineering design. Moreover, the descriptions provided are generally not enough for reproducibility; frequently, the objective functions, decision variable ranges, and constraints are not fully described. In other cases, the optimization requires a simulation module, commonly not easily accessible. In this study, we use the Wind Turbine Design problem, described next.

2.1.9.1 Wind Turbine Design Optimization Problem

It is a problem proposed for the 3rd Evolutionary Computation Competition organized by the Japanese Society of Evolutionary Computation (The Japanese Society of Evolutionary Computation, 2019)[2]. This problem has 32 real coded variables that represent measures and parameters of the turbine. In detail: four variables for blade chord length in different blade span directions; the blade maximum chord length position; four directions of blade mounting angle; five directions of spar-cap thickness; five for trailing edge panel thickness; three for blade precurve; the ratio between rotational speed and wind speed; maximum rotation speed; blade length; tower waist position; three height directions of tower outer diameter and three of tower thickness. The upper and lower bound variate for each variable, normalized, encoded between 0.0 and 1.0, and later converted by the evaluation module.

For the many-objective formulation, there are five objectives. All objectives are of minimization, using negative values for the first objective:

1. Maximize annual power production: to increase profits.
2. Minimize average annual cost: to reduce power generation costs.
3. Minimize tower base load: to reduce construction cost.
4. Minimize blade tip speed: to reduce noise.
5. Minimize fatigue damage: to extend the service life.

Besides, there are 22 constraints, described in the form $g(\mathbf{x}) > 0$. When $g(\mathbf{x})$ is greater than zero, the constraint is satisfied. The constraints prevent damages, such as the collision of the blade with the tower, avoid resonance, and overload. Also, they ensure minimum life, guarantee manufacturability and weldability, and limit noise. Therefore, there are also constraints to prevent failures and avoid non-physical solutions (The Japanese Society of Evolutionary Computation, 2019).

Moreover, the competition provides an evaluation module that computes the objectives and constraints of the decision variables. Besides, it offers a post-processing tool for calculating the hypervolume from the history of solutions found during the search. The maximum number of fitness evaluations (FE) is $10,000$. According to the competition, the module takes about three seconds to evaluate a single solution. Therefore it would take about eight hours to assess all $10,000$ generated solutions (The Japanese Society of Evolutionary Computation, 2019). This number of FE is relatively small. In comparison, a problem with the same number of decision variables in the MaF benchmark methodology would run for $320,000$ FE (Cheng et al., 2017).

2.2 ONLINE SELECTION HYPER-HEURISTICS

As previously discussed, the choice of an algorithm to be applied in an optimization problem is not trivial, especially on many-objective optimization. Hyper-heuristic techniques emerge as high-level approaches to select or generate heuristics automatically. Hyper-heuristic methods aim to find the best heuristic, or the sequence of best heuristics, to be applied to a given problem instead of directly solving it. One of the main motivations is the development of algorithms applicable to a wide range of problems. Therefore, the choice and configuration of an algorithm to be applied would be less arduous (Guizzo et al., 2017).

[2]All information about this problem is available in the competition website: `http://www.jpnsec.org/files/competition2019/EC-Symposium-2019-Competition-English.html`

Hyper-heuristic techniques can be classified into heuristic selection and heuristic generation. In the heuristic generation, the goal is to build, from components or combinations of heuristics, a new heuristic specific to the problem. On the other hand, heuristic selection aims to choose the best heuristics depending on the problem. It is also possible to classify hyper-heuristics according to the learning strategy: online and offline. In offline strategies, learning takes place in a set of training instances. It is expected that the knowledge obtained (and represented in the form of rules or programs) will apply to new instances. On the other hand, the online strategies learn while solving the problem (Burke et al., 2013; Castro et al., 2018).

In this work, we will focus on online heuristic selection strategies. According to Burke et al. (2013), in general, the online heuristic operates in two steps: heuristic selection and move acceptance. In the first, a heuristic is chosen and applied. The simplest method is a random choice. Other selection methods are:

- Choice function based: these selection methods rank the heuristics following (in general) three components, i. intensifying the application of better quality heuristics; ii. quality of the heuristic pair (the objective is to find a cooperative behavior between the heuristic that was applied and the next one); iii. the exploitation of heuristics that have not been used recently. The three components are weighted by weights defined by parameter.

- Multi-Armed Bandit - MAB: this problem considers a set of options from which it is desired to iteratively select which ones will be chosen to maximize the gain obtained. In this case, the choices represent the heuristics. The gain represents the improvement in the solutions obtained when applying that heuristic. An algorithm used for this problem and adapted for the selection of heuristics is the UCB.

- Based on probabilities: for example, the roulette selection method (Castro and Pozo, 2015). In this type of strategy, the selection is stochastic, with a chance for each heuristic. The probabilities are adapted throughout the search, depending on the quality of the heuristics. When heuristics are applied and solutions improve, its probability is increased; otherwise, it is decreased.

The second phase is the move acceptance. After executing the heuristic, the generated solutions can be accepted or rejected. In general, better solutions than the old ones are accepted. Otherwise, an acceptance criterion is applied. The solutions can be accepted or not, depending on the acceptance criteria and parameters. If the solutions are not accepted, the search will proceed from the state before applying the heuristic. Among the existing methods are (Maashi et al., 2015):

- All moves - AM: all solutions are accepted, regardless of their quality is better or worse than the previous state.

- Only Improving - OI: this is the most rigid method, accepting only moves (solutions generated by applying a heuristic) that lead to a better state than the previous one.

- Improving or Equal - IE: in this variation of the previous method, solutions of quality equivalent to the previous state are also accepted.

- Great Deluge: in this strategy, an improved state is always accepted. Otherwise, the solutions are compared to a certain quality threshold. This threshold is predefined and is increased throughout the search. In general, the threshold is initialized with the quality of the initial set of solutions. The increase in the acceptance threshold is also parameter defined. However, finding the most suitable configuration can be difficult.

- Late Acceptance: the quality of the current state is compared to the quality of l iterations ago. The value of l is defined by the parameter.

- Among others, such as Monte Carlo, simulated annealing, and threshold acceptance.

In addition to the heuristic selection and move acceptance methods, another aspect that affects the quality of hyper-heuristics is evaluating the quality of solutions. In single-objective problems in general, quality is assessed by the fitness of the solution generated. In multi-objective optimization, it is necessary to use metrics that evaluate the set of solutions. For example, in Castro and Pozo (2015), is proposed the use of the R2 metric. Other metrics, such as the additive epsilon, could also be used. Also, the scalar function of the metric R2 can be configured, returning different instances of this metric — for example, Tchebycheff, PBI, ASF, or weighted sum.

3 RELATED WORKS

This chapter presents some of the most significant related works on the topics of this research. This review first contains approaches to many-objective optimization. Besides, we introduce and discuss some studies on distributed evolutionary algorithms (dEA). Finally, we discuss the use of hyper-heuristics, especially the ones applied for controlling multiple MOEAs.

3.1 RELATED WORKS ON MANY-OBJECTIVE OPTIMIZATION

A category of algorithms with great prominence in optimization with many objectives is the decomposition-based approach. These approaches demonstrated excellent quality, when applied to the optimization problems of benchmark DTLZ (Deb et al., 2005) and WFG (Huband et al., 2005), ranging from 2 to 15 objectives. Its quality aroused the research and development of several multi-objective evolutionary algorithms based on decomposition (MOEA/D) and algorithms based on reference points (Zhang et al., 2009; Li et al., 2014b; Liu et al., 2014; Deb and Jain, 2014; Wang et al., 2015b; Li et al., 2015c; Castro et al., 2017).

Despite its excellent quality in the benchmark problems, it is worth noting that these algorithms have a drawback. Its quality depends on the shape of the Pareto front and the distribution of weight vectors (or reference points) in space. Recent research exposes and debates this difficulty (Ishibuchi et al., 2017; Li et al., 2018a; Ishibuchi et al., 2020). In those works, the authors assess the dependence between a proper configuration of the weight vectors and the results. Moreover, they expose that none of the evaluated approaches outperform the others in all types of problems. Also, they demonstrate that many-objective optimization is not always difficult for Pareto dominance-based evolutionary algorithms.

For instance, in Ishibuchi et al. (2017), eight algorithms were evaluated: ThetaDEA, NSGA-III, MOEA/DD, NSGA-II, and four variations of MOEA/D using different aggregation functions (PBI, Tchebycheff, Weighted Sum, and IPBI). The algorithms were evaluated in the DTLZ and WFG benchmarks. Weights were distributed using the same methodology presented in Deb and Jain (2014) and (Li et al., 2015c) (we present this method in Section 2.1.4). As noted by Ishibuchi et al. (2017), the distribution of weights using this strategy is compatible with the distribution of solutions on the Pareto front of the WFG and DTLZ problems. In the experiments carried out, the algorithms using this set of weight vectors had the best performances in the WFG and DTLZ problems, mainly the Theta-DEA and MOEA/DD algorithms. Then, inverse problems were developed for the DTLZ and WFG, multiplying the objective values by -1. With the change in the shape of the Pareto front, the quality of those MOEAs has deteriorated. The best results were then obtained by MOEA/D-WS (using weighted sum) and MOEA/D-IPBI (inverted PBI). In these problems, the Pareto dominance-based algorithm, NSGA-II, was not always bad. It was able to achieve better results than Theta-DEA and MOEA/DD in some cases with many objectives. Based on the experimental analysis, the authors concluded that the quality of decomposition-based algorithms depends on the weight vector distribution. Also, performing the proper weight distribution is not a trivial task, as it depends on each problem.

In the Appendix C, we illustrate the presence of the No Free Lunch theorem on many-objective optimization by comparing nine MOEAs. None of them outperformed all others in all problems. Instead, each one was the best in at least one problem instance. As another example, Li et al. (2018a) presents a systematic comparison of 13 algorithms. Those algorithms include several many-objective categories. Despite the advantage of some approaches to several problem instances, the authors conclude that different strategies lead to different search abilities. Moreover, it demonstrates how the hypervolume analysis may not always agree with the observations from the IGD.

Such findings encourage the development of collaborative approaches. In this type of procedure, different algorithms can be applied together. Thus, combining their different strategies to deal with problems of several characteristics. One way to alleviate those issues is through the use of multiple techniques at once. Some hybrid approaches have emerged in the literature to explore collaboration between different algorithms. There are currently some studies on adaptive distributed methods, each with its advantages and disadvantages — for example, A-NSGA-II (H. Jain and K. Deb, 2014) and MOEA/D-AM2M (Liu et al., 2017).

3.2 COOPERATIVE APPROACHES TO MULTI-OBJECTIVE OPTIMIZATION

Some hybrid approaches emerged, trying to combine the strengths of different strategies. Usually, they aim at using different repositories of solutions updated using different criteria. Another way is to decompose an optimization problem and assign different sub-populations. Some hybrid or distributed approaches proposed in the literature are:

- Two _Arch2 (Wang et al., 2015a): this algorithm makes use of two different selection methods used to update two separate repositories. One of the repositories is intended to promote convergence; for that, it uses the Additive Epsilon indicator ($I_{\epsilon+}$). The other repository is intended to support diversity, storing non-dominated solutions. The diversity file is truncated using a diversity maintenance method based on the L_p distance. In the parent selection, one parent is selected at random from the convergence file. In contrast, the other is selected from the diversity file.

- PMEA (Wang et al., 2016): PMEA maintains three different populations, guided by three different environmental selection strategies. The first is based on decomposition, driven by the aggregation function Penalty-based Boundary Intersection (PBI). The second is guided by an indicator ($I_{\epsilon+}$), and the third guided by diversity (Shift-based Density Estimation - SDE). The parent selection is made randomly, with equal probability for each subpopulation. The offspring is generated by SBX crossover and polynomial mutation. Finally, each population is updated with this offspring, each using its own strategy, in parallel (using the master-slave model).

- HEA-DP (Zhang et al., 2016): HEA-DP is a MOEA with two populations that combines decomposition and quality indicator. The environmental selection of the first population uses decomposition, using the PBI aggregation function. The second population uses the $I_{\epsilon+}$ quality indicator for the environmental selection. The parent selection is made randomly, selecting solutions from both populations.

- HEMOA/GD (Li et al., 2015b): HEMOA/GD maintains two populations that evolve collaboratively. One population aims to support diversity, while the other aims to support convergence. The convergence population is guided by grid dominance. While the diversity population is driven by an aggregation function (based on the angular distance between the solution and the weight vector). The parent selection is made by selecting one individual from each population.

- MOEA/D-NL&DE (Wang et al., 2015b): this algorithm uses two different reproduction operators together. Each operator has different characteristics and search abilities. One of the operators is the pair Differential Evolution (Differential Evolution - DE) and polynomial mutation. The other reproduction operator is NL (neighbor learning) and inversion mutation. DE contributes to the global search, while NL promotes convergence. If there is a solution in the neighborhood better than the current solution, the NL is applied. Then the NL uses the best solution in the neighborhood to update the current solution. Otherwise, DE is applied, with a given probability of selecting parents within the neighborhood or the entire population. Most of the works presented above maintain multiple populations (or archives); the MOEA/D-NL&DE demonstrates the cooperation between global and local information.

- MOEA/D-DRA-SPX+CMX (Khan and Zhang, 2010): this work uses two crossover operators in MOEA/D-DRA. The operators used were SPX (Simplex Crossover) and CMX (Center of Mass Crossover). Each operator is applied with a certain probability. The probability is updated adaptively according to the success rate of the operators. The selection of operators works using a roulette wheel, and the weights are adapted during the search.

- NSGA-III-OSD (Bi and Wang, 2016): decomposes the objective space into several subspaces (one for each objective). The parent selection is made in the subpopulation or in the entire population (union of all subpopulations) given a probability. After generating offspring using genetic operators, they are associated with the nearest sub-region. The characteristics of NSGA-III-OSD include that the number of subpopulations is scalable concerning the number of objectives. Also, the division of space is simple and independent of the number and distribution of weight vectors. Finally, each subpopulation is guided towards a different region of the search space.

- MOEA/D-M2M (Liu et al., 2014), which decomposes the objective space into several constrained subproblems. Each subpopulation runs an NSGA-II and generates offspring by selecting parents only from its population. Also, each of the constrained sub-problems is simpler than the original problem. Finally, the optimization of all subproblems is equivalent to the optimization of the original problem. The most notable difficulty is to decompose the space into a pre-defined number of equally distributed sub-regions

- MOEA/D-DI (He and Yen, 2014), which uses an independent population for each objective. In the first step of the algorithm, the goal is to find the extreme values for each objective. The search is guided by the ASF function, and each subpopulation is associated with one objective. After a certain number of generations, the best solution for each objective is identified. In the second step, the search is guided by diversity, using a single population. The algorithm was evaluated on the set of DTLZ test problems. However, their conclusions may be biased towards the problem set used. Some DTLZ problems have the characteristics that one group of variables is responsible for diversity. In contrast, another group is responsible for convergence. Therefore, convergence and diversity could be easily solved separately, which is not valid on most real-world problems (Ishibuchi et al., 2017).

Some of those research make use of multiple approaches to solve a many-objective problem. Others use the same strategy but decompose the problem instance into smaller problems. Those research has inspired a preliminary study on the contribution of different techniques and the exchange of information. Moreover, the most significant difference between this thesis and prior related works include the use of hyper-heuristics, the application of several different MOEAs; and the proposed migration procedure to exchange solutions among MOEAs.

3.2.1 Distributed Evolutionary Algorithms

The distributed evolutionary algorithms (dEA) for multi-objective optimization is a hot-spot in the dEA field (Gong et al., 2015). The multi-objective search is facilitated when using distribution, maintaining the population diversity, and avoiding local optima. The planning of a distributed project has some design choices to be made, for example, the model of distribution, such as master-slave, island, or cellular. Another decision is if the processes will be homogeneous or heterogeneous. If homogeneous, all processes execute the same algorithm with the same configurations. If heterogeneous, the processes run different configurations or even different algorithms. In the case of Heterogeneous approaches, there is the decision of which algorithm/configuration to execute on each process. There is also the granularity level: population, individual, operator, or variable. Finally, there is the decision whether the process communication will be synchronous or asynchronous. On asynchronous communication, the process shares its information whenever it decides to, independently from the others. In synchronous communication, all processes must communicate at the same moment. Another major decision for distributed projects is the communication rules, such as the frequency, the direction, and the content. It is necessary to decide when to send information to the other processes, which communication topology to use, which information to send, and what to do with the received information (Gong et al., 2015). The study on distributed evolutionary algorithms for multi-objective optimization were the initial steps for this research. A heterogeneous island model was proposed, being evaluated for both synchronous and asynchronous communication. Later, the hyper-heuristic online selection was incorporated in the study.

3.3 HEURISTIC SELECTION FOR CONTROLLING MULTIPLE MOEAS

There are in the literature some works on multi-objective hyper-heuristics (MOHHs). Most of them performing Adaptive Operator Selection (AOS) (Gonçalves et al., 2018; Castro et al., 2018; Guizzo et al., 2017; Gonçalves et al., 2017; Li et al., 2014a). Just a few of them make use of different MOEAs. In this research, we focus on the heuristic selection for controlling multiple metaheuristics (Drake et al., 2020). From those, it is possible to divide into three categories: single population dynamic resource allocation (DRA), multiple population DRA, and heuristic selection.

1. In *single population dynamic resource allocation*, we can cite the Multi-Indicator Genetic Algorithm (MIGA) (Vázquez-Rodríguez and Petrovic, 2013) AMALGAM (Vrugt and Robinson, 2007) and MOABHH (de Carvalho and Sichman, 2018) based on Copeland voting. Those works execute all MOEAs every iteration. Every MOEA receives a share of the population, according to its share, of size δ. Then, each one generates δ solutions and updates back to the population. The number of solutions δ produced is adaptively updated, given the contribution of the MOEA to the population. In other words, the hyper-heuristic adaptively divides the population (the better the performance of a MOEA on the past generation, the higher is the number of individuals of the population share it will receive).

2. On *multiple populations dynamic resource allocation*, as EF-PD (Wang et al., 2018), every MOEA has an internal population of size s. At each iteration, each MOEA generates δ new solutions based on its population. Then, it sends all newly created solutions to an external repository. Finally, every MOEA updates its internal population based on the external repository of newly generated solutions. Then, the value δ of every MOEA is adaptively updated, given its contribution to the external repository.

3. In hyper-heuristics based on *heuristic selection*, one MOEA is selected and executed, receiving the population as input. This population is then updated by the MOEA using its parent selection, reproduction, and environmental selection strategies. Afterward, the new population is returned to be evaluated and to compute the reward of the applied MOEA. After that, the external population is replaced by the new population, using an acceptance criterion. Then, the high-level approach uses the reward information to update its selection criterion. Finally, a new heuristic is selected, receiving the external population as input. This process is repeated until the termination criterion is satisfied. One example of *heuristic selection* is the Choice-Function based hyper-heuristic CF-HH (Maashi et al., 2014, 2015). In those papers, CF-HH online selects from three MOEAs — it outperformed the three MOEAs and AMALGAM. Another example is HH-LA (Li et al., 2018b), based on Learning Automata — HH-LA achieved better results than CF-HH.

3.4 DISCUSSIONS

The related works encourage research on the cooperation of multiple MOEAs for many-objective optimization. Moreover, this literature review highlights the importance of using test problems with different characteristics. For example, Ishibuchi et al. (2017) concludes that the performance of the MOEAs is dependent on the instance being optimized. It is also crucial that the set of MOEAs comprises different characteristics. For example, some related works demonstrate that collaboration between different approaches can improve the performance of the search (Wang et al., 2015a, 2016; Zhang et al., 2016; Li et al., 2015c).

Moreover, the research on hybrid algorithms is suggested by Li et al. (2015a), combining two or more approaches. The approach proposed in this research is a way of achieving that hybridization. Most of the related works share with this proposal the characteristic of combining different strategies. However, for most works cited above, only methods of environmental selection or reproduction are combined. Whereas the proposed approach aims to combine complete MOEAs. Finally, the proposed framework makes use of multi-objective hyper-heuristics (MOHH).

Despite the performance of MOHH for two and three objectives, when increasing this number of objectives, they face new challenges. First, the pool of MOEAs for solving Many-objective Problems (MaOPs) should include a diversity of characteristics. Furthermore, the state-of-the-art MOEAs for MaOP usually keep other information about the evolutionary state besides the population, e.g., an estimated nadir point, an ideal point, an archive of solutions. However, state-of-the-art MOHH does not include an easy way of keeping this information. When the population is external, the MOEA may lose solutions that are important according to its criterion. For example, suppose that MOEA/DD was selected and initialized with the external population. Then, it returned an output, including a dominated solution that is considered necessary for diversity. Suppose that the next applied MOEA is NSGA-II, which the primary criterion is Pareto dominance. Consequently, the dominated solution will not survive until the next time that MOEA/DD is applied. For more information about multi-objective selection hyper-heuristics, see Section 6 of Drake et al. (2020).

4 COOPERATION OF MOEAS FOR MANY-OBJECTIVE OPTIMIZATION

In this work, we propose the cooperation of multiple MOEAs for many-objective optimization. First, we studied the use of distributed MOEAs, running simultaneously during the search, exchanging information, and proposed HeDi (see Section 4.3). Next, to weight the participation of each MOEA on the search, we introduce the use of hyper-heuristics. Thus, we proposed a preliminary hyper-heuristic approach as a proof of concept called HHcMOEA (Section 4.4).

The HHcMOEA uses hyper-heuristics to the cooperation of multiple MOEAs to solve many-objective problems. Besides, we incorporate a migration step in the online selection hyper-heuristic framework. This migration step is not present in other hyper-heuristic approaches. The goal is exchanging information among the MOEAs, to improve the overall search ability. This procedure allows MOEAs to keep their knowledge updated to the current search state, even if they have not been applied recently. Moreover, since each MOEA has its criteria to select the solutions that will be preserved for the next generation, in this work, every MOEA has its internal population. This behavior is particularly suitable for many-objective optimization. This approach was used to demonstrate the importance of the proposed migration procedure. Then, this hyper-heuristic was improved, given the knowledge from the preliminary analysis and further experiments. It resulted in a newly proposed framework, HH-CO. The HH-CO represents a feasible strategy for optimizing many-objective problems, joining strengths, and overcoming the limitations that MOEAs have separately.

4.1 MIGRATION PROCEDURE

A crucial concept of this research is the exchange of information. The goal is to keep the knowledge of every MOEA updated at every iteration. When a MOEA executes, it shares solutions to the other MOEAs. The migration is made using the environmental selection method of every MOEA. Therefore, every MOEA keeps a population of solutions, filtered by its criteria. This procedure prevents a MOEA from losing solutions that it considers necessary, but another MOEA would discard it. Furthermore, during this phase, the MOEAs update their internal information, for example, an estimation of the nadir point. Therefore, the next time the MOEA is executed, its population and internal information are up to date concerning the search state and the external information received.

On the first proposed approach (HeDi — see Section 4.3), all MOEAs are applied and exchange information every iteration. The main idea is to incorporate diversity by exchanging solutions among different MOEAs. On the next proposed approaches (HHcMOEA and HH-CO — see sections 4.4 and 4.5) only one MOEA is applied at a time, and exchange information with all other MOEAs. We use hyper-heuristics to select which MOEA will be applied next, and the use of the proposed migration step allows them to keep their internal information updated, even when they are not being applied. Besides, it will enable a MOEA to benefit from solutions generated by others and improve its search.

For example, suppose that we have a pool of two MOEAs: NSGA-II and MOEA/D and consider that NSGA-II was applied. In the proposed approach, solutions from NSGA-II will be sent to the environmental selection step of MOEA/D. It will associate each solution to a subproblem and then update its internal population using its strategy. When MOEA/D is applied, it sends solutions to NSGA-II. Then, NSGA-II incorporates them based on Pareto dominance. The NSGA-II and MOEA/D population are kept updated during the entire search, using different criteria. Differently, on other hyper-heuristic approaches, the population of NSGA-II would be set as the initial population for MOEA/D. And later, the population updated by MOEA/D would be set as the initial population of NSGA-II. Therefore, the information gained by one criterion may be lost when a different criterion is applied. In this study, the proposed approach uses broadcast topology, i.e., every MOEA communicates to every other. However, it can handle any neighborhood topology, such as ring topology or an adaptive topology, since every MOEA keeps a list of neighbors. It is worth noticing that it is a logical topology, as the algorithms are placed physically on the same thread.

4.2 POOL OF MOEAS

In this study, we make use of multiple complete multi-objective evolutionary algorithms (MOEA) simultaneously. Every MOEA in the pool implements parent selection, reproduction, and environmental selection, following its strategy without modification. We have used a set of ten MOEAs[1] composed of different characteristics through this research. The goal is to create a search ability capable of dealing with the difficulties posed by various many-objective problems. We have used five decomposition-based algorithms: MOEA/D (Zhang and Li, 2007); MOEA/D-STM (Li et al., 2014b); NSGA-III (Deb and Jain, 2014); MOEA/DD (Li et al., 2015c); and ThetaDEA (Yuan et al., 2016). The pool also includes two different Pareto based algorithms, NSGA-II (Deb et al., 2002) and SPEA2 (Zitzler et al., 2001). Depending on the problem instance characteristics, Pareto based approaches may achieve better results than state-of-the-art MOEAs, even when the number of objectives is ten or more (Ishibuchi et al., 2017). Moreover, the pool includes two indicator-based MOEAs: MOMBI2 (Gómez and Coello, 2015), based on R2; and HypE (Bader and Zitzler, 2011) based on hypervolume estimation. Additionally, it incorporates SPEA2SDE (Li et al., 2014c) that modifies the diversity mechanism of SPEA2.

All experiments in this thesis use the jMetal framework[2]. The experiments for HeDi uses the NSGA-III from jMetal. However, the experiments for HHcMOEA and HH-CO uses the implementation from ManyEAs repository[3], same as ThetaDEA. We opted for this implementation after analyzing that it produces better results, which was more compatible with the NSGA-III literature (see Appendix A.2). Moreover, SPEA2SDE and HypE are not present in the preliminary analysis since a compatible implementation was not available at the time. A jMetal implementation of HypE and SPEA2SDE was later provided by Dr. Yuan Yuan by e-mail and included in the experiments. Finally, the MOEA/D-STM implementation is available online by the author, Dr. Ke Li[4]. However, after the first experiments, we stopped using this algorithm due to its computational cost for many-objective optimization. All other MOEAs are natively available in the jMetal framework.

4.3 HETEROGENEOUS DISTRIBUTED APPROACH: HEDI

This section describes the proposed distributed framework for the cooperation of many-objective evolutionary algorithms (HeDi). This framework results from our initial effort on the cooperation of multiple MOEAs, which did not include the use of hyper-heuristics. The framework uses

[1]Some state-of-the-art MOEAs were not included due to their source code being not compatible with the jMetal framework used. Some examples are CVEA3, AMPDEA, and BCE-IBEA.

[2]For information about the jMetal framework see (`https://github.com/jMetal/jMetal`)

[3]ManyEAs, provided by Dr. Yuan Yuan on its GitHub repository (`https://github.com/yyxhdy/ManyEAs`)

[4]MOEA/D-STM implementation repository at `https://github.com/JerryI00/releasing-codes-java`

different algorithms to explore the search space. The idea behind this framework is that an algorithm can present features that can help the search in some regions of the search space. In contrast, in other regions, the algorithm could be trapped in local optima. However, the cooperation with an algorithm with a different strategy can help escape from this situation. As a result, the framework has a better search ability.

The framework uses the population distributed in an island model because research reports that it helps to avoid premature convergence (Gong et al., 2015). The framework instantiates several islands, each island with a small population size and executing a specific MOEA. Besides, each island has migration criteria, a selection policy for the information that will be shared, and a guideline for the information received from another island. The information obtained from another island is kept in a queue until the receiver MOEA is ready to handle them.

In the HeDi framework, first, all islands are initialized. Then all islands execute simultaneously, each one following the steps of its specific MOEA. Every iteration, the island sends solutions to all other MOEAs and consumes all solutions received. The topology of the island model is locally managed, as each MOEA maintains a list of neighbors. The framework implements two strategies of communication: synchronous and asynchronous. If the execution of the processes is synchronous, the islands must wait for the others before starting the communication. When all islands meet the stop criterion, all populations are joined. The output is the non-dominated set of solutions.

Algorithm 2 presents the procedure executed for each island in the distributed cooperation. After the initialization of every island, the MOEA applies its strategy for selection, reproduction, and replacement until it reaches a stopping condition. Every iteration, the island selects the information to be sent to the neighbor islands and then sends it. The information transmitted is the offspring generated during that iteration. Next, the island consumes all its queue, containing the solutions received from its neighbors. It combines the current population, the offspring generated by itself, and the solutions from its queue received from other MOEAs. Next, the island applies the environmental selection step considering this combined set of solutions.

Algorithm 2: Procedure executed by each island in the proposed heterogeneous distributed approach.

```
Data: MOEA to be executed
initialization;
while the stop criteria is not met do
    /* each step is executed according to each MOEA    */
    selection;
    reproduction;
    fitness_evaluation;
    select solutions for migration;
    send the selected solutions;
    if synchronous approach then
        /* synchronization barrier                      */
        wait;
    end
    consume solutions from neighbors;
    environmental selection;
end
```

The general framework proposed can be implemented in several ways, as each design choice leads to a different implementation. In this research, we implemented the proposed framework as follows: The migration occurs every iteration; the MOEA sends to the others all offspring generated on that iteration. The HeDi uses a complete graph topology sharing the information among all islands. Moreover, we built two different implementations for the synchronous and asynchronous versions. On the synchronous version, all algorithms run on the same thread, one at a time. When all of them have finished one iteration, then it is applied migration. This implementation is more straightforward than synchronizing threads and leads to the same results regarding the quality of the approximation fronts generated. However, if execution time was taken into account, a parallel implementation would have been more efficient.

In this work, we do not focus on the execution time of HeDi, but how we can use the distribution to integrate information among different algorithms. Moreover, the term distributed on synchronous HeDi means that the information is *distributed* on different algorithms (on the same thread/machine). On the asynchronous implementation, we use Java Threads, and the information is shared through a queue that handles the concurrency. Moreover, each threat (running one algorithm) knows the other MOEAs queue and uses those queues for migration. On asynchronous HeDi the information is distributed on different threads but still on the same machine. In other words, the distributed term in HeDi does not refer to *distributed computation* where information is shared among multiple computers.

The analysis of the proposed framework is presented in Section 5.2. We evaluate both synchronous and asynchronous communication, instantiated for two MOEAs. The good results obtained demonstrated the importance of the proposed migration step. However, the results of this initial framework did not present scalability to increase the number of MOEAs (see the synchronous HeDi further studies at Appendix A). Those issues, identified by extensive experimentation, guided us towards the use of hyper-heuristics. Initial proof of concepts to analyze the use of hyper-heuristics is presented in Appendix B.

4.4 PRELIMINARY HYPER-HEURISTIC APPROACH: HHCMOEA

Initially, we proposed a hyper-heuristic framework called HHcMOEA. This framework differs from other hyper-heuristic frameworks, mainly by the inclusion of the proposed migration step. Therefore, it was used to demonstrate the importance of the proposed information exchange method. The proposed HHcMOEA is presented at Algorithm 3. First, all MOEAs are initialized, each one with its own population and parameter values. The hyper-heuristic parameters are also initialized. Then, the following steps are executed until the stop criterion is met. First, one MOEA is selected using a heuristic selection method. Next, the current population of this MOEA is copied (pop_{old}). Then, the selected MOEA is executed for a given number of iterations. The reward of the selected MOEA is given, comparing the quality of the population before and after the execution of the MOEA. Finally, the new population is shared with the neighborhood of the executed MOEA.

The proposed algorithm incorporates the proposed migration phase. It is worth noticing that, in HHcMOEA, the executed MOEA sends its current population to neighbors. In other words, it sends its population after the environmental selection step is applied. Moreover, each iteration, the HHcMOEA selects and applies a MOEA. The MOEAs are selected given a heuristic selection method. This method considers the reward that the MOEA received each time it was

Algorithm 3: Preliminary approach: HHcMOEA

Data: pool of MOEAS

initialization;

while *the stop criterion is not met* **do**

    $selected \leftarrow$ heuristic_selection(MOEAS);

    $pop_{old} \leftarrow$ copy_population_from($selected$);

    $pop_{new} \leftarrow$ execute($selected$);

    $metrics \leftarrow$ extract_metrics(pop_{old}, pop_{new});

    set_reward($selected$, $metrics$);

    foreach $moea \in$ *get_neighborhood($selected$)* **do**

        migration($moea$, pop_{new});

    end

end

applied. This reward is computed as the fitness improvement rate of a given quality indicator $I(pop)$:

$$reward(pop_{old}, pop_{new}) = \frac{I(pop_{old}) - I(pop_{new})}{I(pop_{new})} \tag{4.1}$$

Also, the HHcMOEA uses the roulette based selection method proposed by Castro et al. (2018). In this method, all heuristics have the same initial probabilities of being selected. The heuristic selection method randomly chooses a MOEA based on those probabilities. Depending on the reward, the probability of the selected MOEA is incremented or decremented by a fixed amount. A minimal probability value is kept avoiding that any MOEA has zero probability of being selected.

In the experimental analysis (Chapter 5), we compare HHcMOEA to a version without the information exchange (HHMOEA). In HHMOEA, the initial population of the next selected MOEA is the population from the previous one. The knowledge achieved in this analysis motivated further research and experimentation that led to the development of the HH-CO presented in the next section.

4.5 IMPROVED HYPER-HEURISTIC APPROACH: HH-CO

In this section, we describe the proposed Cooperative based Hyper-heuristic (HH-CO). It has three major characteristics (Figure 4.1):

1. Every iteration, it selects a MOEA based on its improvement in the last iteration.
2. Each MOEA has an internal population and uses this population to generate offspring.
3. The MOEA shares its offspring after being applied, and the other MOEAs may incorporate solutions into their own population.

Algorithm 4 presents the proposed HH-CO framework. First, it initializes the MOEAs with randomly generated populations. Then, HH-CO repeats the following steps until the termination criterion is satisfied. Initially, it copies the population of each MOEA. Then, the selection criteria return the next MOEA to be applied. The selected MOEA is executed and produces newly generated solutions (offspring). After that, it sends the offspring to every MOEA

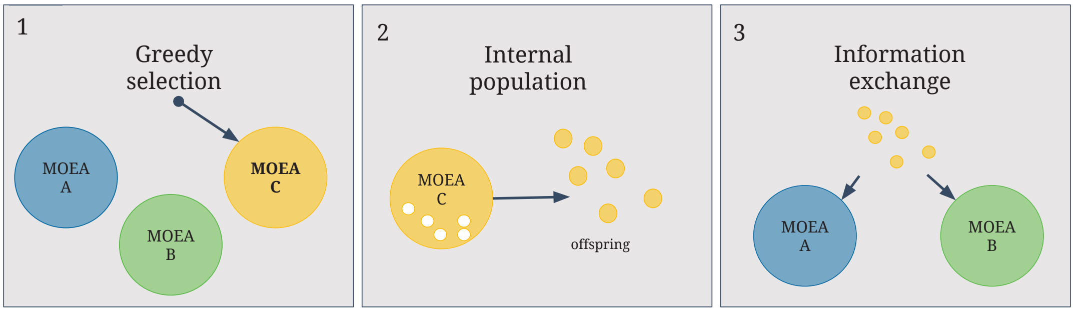


Figure 4.1: Cooperative based hyper-heuristic characteristics

(migration procedure). Then, each MOEA is rewarded by evaluating the improvement of its population before and after the environmental selection step. Finally, every MOEA receives a reward. When the iterative process is finished, the output of HH-CO is the union of all populations, removed dominated solutions and, truncated by the population size. Details of the exchange migration procedure and reward are given in Section 4.1. Section 4.2 presents the MOEAs that compose the pool.

Algorithm 4: Improved approach: HH-CO

```
Data: pool of MOEAs
initialization;
while the stop criterion is not met do
    foreach moea ∈ MOEAs do
        old_i ← copy_population(moea);
    end
    selected ← heuristic_selection(MOEAs);
    offspring ← execute(selected);
    foreach moea ∈ MOEAs do
        migration(moea, offspring);
        new_i ← copy_population(moea);
        reward ← get_improvement(old_i, new_i);
        set_reward(moea, reward);
    end
end
```

A crucial point in HH-CO is that each MOEA keeps a list of solutions generated during the iteration (offspring). A copy of this list is what is sent when the migration procedure is called. Therefore, in the HH-CO, the other MOEAs may incorporate even solutions that the applied MOEA would have discarded. A second crucial point of HH-CO is the credit assignment. In HH-CO, all MOEAs receive a reward at every iteration. The reward is computed by the improvement of its population (similar to HHcMOEA, see Equation 4.1). In other words, we calculate the R2 of the population before and after the current iteration and measure the improvement.

The applied MOEA has its population modified by environmental selection. In contrast, the other MOEAs have their population modified by the migration procedure (which uses the environmental selection). In HH-CO, the heuristic selection procedure is greedy (with no second criteria). In this way, since HH-CO evaluates the improvement of the population on the current iteration, it selects the MOEA with the most significant improvement in the last iteration. HH-CO looks for MOEAs that are improving the search instead of stagnated ones. However, due to the exchange of information, stagnated MOEAs may improve later in the search.

4.5.1 Differences to HHcMOEA

The HH-CO presents two significant improvements compared to HHcMOEA. First, the information exchange method of HH-CO sends all newly generated solutions (offspring). On the other hand, HHcMOEA shared the population updated (after the environmental selection method of the applied MOEA). Second, in HH-CO, all MOEAs are rewarded every iteration, while only the applied one was rewarded on HHcMOEA. We changed this behavior since all MOEAs are updated (from migration). Therefore all of them have a chance to improve. Finally, the selection method of HHcMOEA was a roulette wheel. In this way, it could guarantee a minimal chance of each MOEA being applied. The HH-CO uses a greedy selection since all MOEAs receive solutions and are evaluated every iteration. Therefore it already assures a minimal change of each one being applied. In this way, the selection used by HH-CO favors MOEAs that are improving instead of stagnated MOEAs. However, stagnated MOEAs may improve later due to the information exchange. Moreover, the pool of seven MOEAs from HHcMOEA was extended to include SPEA2SDE and HypE, improving the diversity from the pool and the quality of the results.

4.5.2 Discussions

In the proposed approach, every MOEA in the pool has its population. When a MOEA is selected, it generates new solutions to update its population. Then, it shares information with the other MOEAs. In this step, every MOEA accepts external solutions according to its criterion. All MOEAs update their internal information during the evolutionary process. Moreover, compared to MOHHs with subpopulations, the main differences are: in the proposed framework, only one MOEA is executed at a time; every MOEA has an entire population. The population size is fixed during the whole search.

The generality of the proposed approach allows using different types of MOEAs, for example, an indicator based, Pareto-based, or decomposition-based. Also, it supports the incorporation of other strategies such as MOPSOs (multi-objective particle swarm optimization) and MOEDAs (multi-objective estimation of distribution algorithms) alongside the MOEAs. Moreover, it allows the use of steady-state algorithms (e.g. MOEA/DD). The only modification applied in those MOEAs is that the solutions generated in one iteration must be kept in a list. Furthermore, it focuses on many-objective optimization. This kind of problem requires different approaches than the ones used by MOHHs applied to problems with two or three objectives. (Li et al., 2018b). Finally, the scalability level of the proposed framework allows the use of a large number of MOEAs together. The next chapter details the empirical evaluation of the proposed approaches.

5 EMPIRICAL EVALUATION

In this chapter, we first evaluate the cooperation of MOEAs using a proposed distributed framework (HeDi). We assess both synchronous and asynchronous communication. Next, we analyze the impact of the migration procedure on a proposed hyper-heuristic framework (HHcMOEA). This proof of concept instigated the development of an improved version, the HH-CO. The good results of HH-CO compared to a state-of-the-art hyper-heuristic motivated the analysis of a real-world problem. Therefore, the HH-CO is evaluated on the Wind Turbine Design problem. Its results are compared to the MOEAs that compose its pool.

5.1 EMPIRICAL ANALYSIS SETUP

In this research, we have used ten MOEAs, namely: NSGA-II, SPEA2, ThetaDEA, NSGA-III, MOMBI2, SPEA2SDE, HypE, MOEA/D, MOEA/D-STM, and MOEA/DD. However, the MOEAs used in different experiments vary since MOEAs were gradually included or removed as this research evolved. We selected those state-of-art algorithms due to their diversity of characteristics and good results presented in the literature. Including Pareto based MOEAs, since recent research has demonstrated that they can outperform many-objective EAs depending on the problem characteristics. This includes even problems with a high number of objectives (e.g., 10) (Ishibuchi et al., 2017). In these analyses, the parameter configuration of each MOEA was set based on the literature of each algorithm. Therefore, simulating an off-the-shelf use, rather than a fine-tuning to find the best parameter setting for each algorithm. The discussions are drawn over the different overall strategies used by each MOEA. Finally, the parameter setting of each MOEA is the same when applied individually or within the hyper-heuristic. All of them use Polynomial mutation, with a probability of $1.0/n$, where n is the number of decision variables and mutation index of 20.0. The crossover operator of MOEA/DD, ThetaDEA, and NSGA-III is SBX with a 1.0 probability and crossover distribution index of $\eta_c = 30.0$ (Li et al., 2015c; Deb and Jain, 2014; Yuan et al., 2016). SPEA2SDE and HypE also use SBX, with 1.0 probability and $\eta_c = 20.0$ (Li et al., 2014c; Bader and Zitzler, 2011). MOMBI2, SPEA2, and NSGA-II use SBX as well, with 0.9 probability and $\eta_c = 20.0$ (Gómez and Coello, 2015; Zitzler et al., 2001; Deb et al., 2002). On the other hand, MOEA/D uses rand/1/bin Differential Evolution (DE), with $CR = 1.0$ and $F = 0.5$ (Zhang and Li, 2007). Other parameters include ThetaDEA penalty parameter $\theta = 5.0$ (Yuan et al., 2016) and HypE sample size of 10000 (Bader and Zitzler, 2011). For the MOEA/D-STM, the parameter configuration is the same reported in (Li et al., 2014b). The reproduction is performed using differential evolution (DE) and polynomial mutation. The mutation probability is $p_m = 1/n$ and the distribution index is $\eta_m = 20$. The DE parameters CR and F are set to $CR = 1.0$ and $F = 0.5$. The neighborhood size T was set to 20 and the probability of selecting from the neighborhood of $\delta = 0.9$.

In the experimental analysis, the hyper-heuristics uses the improvement rate of the R2 quality indicator to evaluate the quality of the application of a MOEA. The improvement rate is computed using the R2 value of the selected MOEA population before and after its application. The R2 parameters used were: Tchebycheff scalarization and the same weight vectors used by the decomposition-based MOEAs (Deb and Jain, 2014; Li et al., 2015c; Brockhoff et al., 2012). The R2 has a lower computational cost than hypervolume, with a correlated behavior (Brockhoff et al., 2012).

For each experiment, we present a per problem analysis. This analysis is presented by tables with the average values of each indicator. In this analysis, the best value *for each problem instance* is displayed in bold. The values statistically equivalent to the best are presented with a gray background. This statistical comparison was made per problem, using the Kruskal Wallis test, with 95% significance. It is shown pairwise, using the Tukey and Kramer post hoc test with 95% significance. Additionally, the raw value of the independent runs was used as input for this statistical test. This analysis presents the behavior of each algorithm on different problem instances.

Also, for some analysis, we make use of the critical difference plot. This plot connects the algorithms to their average ranking. A bold horizontal line connecting two (or more) algorithms represents that the algorithms did not achieve a statistically significant difference from one another. This overall statistical evaluation is pairwise performed by the Nemenyi test, with 95% significance (Demšar, 2006). The input was composed of the average value of each problem instance. This analysis presents the overall behavior of each evaluated algorithm.

Moreover, for some analysis, we make use of boxplots. The boxplot allows visualizing the distribution of data through their quartiles. A box extends from the first to the third quartile (the interquartile range - IQR) and measures variability, with a line at the median. Circles represent outliers, i.e., data points distant from the box more than $1.5 \times IQR$ (Wickham and Stryjewski, 2012). Finally, it is worth noticing that except AsyncHeDi, all other algorithms are executed in single-thread experiments. The parallel implementation of the proposed approaches, mainly the migration step on HH-CO, is left for future works. This affects the execution time but has no effect in the results of the algorithms in terms of convergence and diversity.

5.2 EMPIRICAL ANALYSIS OF THE DISTRIBUTED COOPERATION OF MOEAS

This section presents the experimental setup used for the empirical validation of the HeDi framework. The goals of this validation are (i) evaluation of the hypothesis that the cooperation between algorithms can improve the results of the MOEAs working alone; and (ii) the comparison of both synchronous and asynchronous communication. As detailed in Section 4.3, only the asynchronous version of HeDi is multi-thread. In contrast, the synchronous version is executed in a single thread. However, we do not account for computation time. We are interested only in how the information exchange and the cooperation of multiple MOEAs can lead to *better results* in terms of convergence and diversity to the Pareto Front.

In the experimental analysis, four algorithms are used. 1) The Asynchronous version of the cooperation between NSGA-III and MOEA/D-STM, using the proposed approach (AsyncHeDi); 2) The Synchronous version SyncHedi; 3) The NSGA-III and; 4) the MOEA/D-STM. For that comparison, six benchmark problems were selected: the DTLZ1 to DTLZ4, from DTLZ (Deb et al., 2005), WFG6, and WFG7 from WFG (Huband et al., 2005). This is the same benchmark used in related works for many-objective optimization (Deb and Jain, 2014; Li et al., 2015c). The experiments were performed using 3 to 15 objectives (m). The selected benchmark problems represent different characteristics of MOPs (Multi-Objective Problems), such as linear and concave shapes, multi-modality, and separability. For instance, the DTLZ1 and DTLZ3 are hard-to-converge multi-modal problems, WFG6 is a non-separable reduced problem, and WFG7 is separable, unimodal (Wang et al., 2015a). Those benchmark problems were selected due to their extensive application on the literature (Deb and Jain, 2014; Li et al., 2015c).

The methodology used for the experiments is based on (Deb and Jain, 2014). The stop criterion is determined by the number of iterations. It is set for different test instances and variates per problem and the number of objectives (presented at Table 5.1). Moreover, each algorithm runs for 20 independent runs (Deb and Jain, 2014).

Table 5.1: Number of iterations for different test instances.

	Number of objectives (m)				
Problem	3	5	8	10	15
DTLZ1	400	600	750	1000	1500
DTLZ2	250	350	500	750	1000
DTLZ3	1000	1000	1000	1500	2000
DTLZ4	600	1000	1250	2000	3000
WFG6	400	750	1500	2000	3000
WFG7	400	750	1500	2000	3000

Both MOEA/D-STM (Li et al., 2014b) and NSGA-III (Deb and Jain, 2014) (described in Section 2.1.3) use a reference set of points to guide the search. The reference set of points was split between the algorithms, alternating between one and the other. We configure the population size of each island according to the number of reference points associated with it. Table 5.2 presents the population size (and the number of reference points) associated with each algorithm. The reference points and the population size are based on (Deb and Jain, 2014). IGD and Hypervolume (Li et al., 2015c) were used to evaluate the non-dominated sets returned for every algorithm (see Section 2.1.6). As pointed out by Li et al. (2015c), it is possible to generate a set of uniformly distributed points on the Pareto front for the problems addressed in the present section. This generation is viable because the Pareto-optimal surfaces of the problems are known a priori. It is possible to generate a set based on the intersection of the vectors with the Pareto-optimal surface using a set of weight vectors (reference points) (Li et al., 2015c). The set of weight vectors were generated using Das and Dennis's approach (Deb and Jain, 2014). And the reference points to the computation of IGD were generated using the strategy presented in by (Li et al., 2015c). For the hypervolume, in this analysis, we set the nadir point for each problem instance, and any solution worse than the nadir point, in any objective, was not considered for the hypervolume computation (Li et al., 2015c). The nadir point for DTLZ1 was set to $(1.0, \ldots, 1.0)$; for DTLZ2 to DTLZ4 it was set to $(2.0, \ldots, 2.0)$; and for WFG6 and WFG7 it was set to $(3.0, \ldots, 2.0 \times m + 1.0)$ (Li et al., 2015c).

Table 5.2: Population size (and reference points) used by the algorithms.

m	HeDi	NSGA-III	MOEA/D-STM
3	93 (91)	92 (91)	91 (91)
5	213 (210)	210 (210)	210 (210)
8	158 (156)	156 (156)	156 (156)
10	277 (275)	276 (275)	275 (275)
15	135 (135)	136 (135)	135 (135)

5.2.1 Experimental results

This section presents the comparison among four algorithms: the synchronous (SyncHeDi) and the asynchronous (AsyncHeDi) cooperation, NSGA-III, and MOEA/D-STM. Tables 5.3 and

5.4 present the results of mean values of Hypervolume and IGD. From the IGD results, it is possible to observe that the Synchronous version of the Heterogeneous Distributed approach (SyncHeDi) achieved the best or had no statistically significant difference to the best average result in all 30 problems. In 21 of them with a statistically significant difference from the MOEAs applied alone; and statistically different to the Asynchronous version in 12 problem instances. The Asynchronous version (AsyncHeDi) was the best, or equivalent to the best, in 18 problem instances. In 12 problem instances, it was better than the MOEAs applied alone, with statistical significance. When analyzed the MOEAs applied alone, the MOEA/D-STM was the best or equivalent to the best in 9 problem instances, 3 of them better than the AsyncHeDi. In one problem, the NSGA-III was the best but equivalent to all others.

Table 5.3: The mean of IGD. The best results are in boldface and a statistical equivalence to the best in a gray background.

		HeDi		MOEAs applied alone	
Obj.	problem	Async	Sync	MOEA/D-STM	NSGA-III
3	DTLZ1	8.03E-3	**7.17E-3**	8.15E-3	2.25E-2
	DTLZ2	9.77E-3	8.98E-3	**8.97E-3**	1.95E-2
	DTLZ3	1.07E-2	**9E-3**	9.17E-3	3.5E-2
	DTLZ4	1.25E-2	**1.23E-2**	1.49E-2	3.67E-2
	WFG6	**1.08E-2**	1.09E-2	1.2E-2	2.37E-2
	WFG7	1.04E-2	**9.6E-3**	1.03E-2	3.83E-2
5	DTLZ1	1.48E-2	**1.07E-2**	2.19E-2	2.12E-2
	DTLZ2	1.81E-2	**1.57E-2**	2.21E-2	2.11E-2
	DTLZ3	1.64E-1	**1.59E-2**	1.51E-1	5.92E-2
	DTLZ4	**1.85E-2**	1.88E-2	2.38E-2	3.48E-2
	WFG6	1.99E-2	**1.68E-2**	2.38E-2	3.07E-2
	WFG7	2.75E-2	**2.1E-2**	3.31E-2	3.54E-2
8	DTLZ1	3.36E-2	**2.88E-2**	3.26E-2	8.73E-2
	DTLZ2	4.84E-2	**4.57E-2**	5.39E-2	5.76E-2
	DTLZ3	4.25E-1	**4.66E-2**	5.6E-2	5.37E0
	DTLZ4	4.8E-2	**4.49E-2**	5.96E-2	8.34E-2
	WFG6	**3.99E-2**	4.1E-2	4.44E-2	6.35E-2
	WFG7	6.1E-2	6.13E-2	**5.87E-2**	7.8E-2
10	DTLZ1	2.61E-2	**2.36E-2**	2.72E-2	6.84E-2
	DTLZ2	4.05E-2	**3.95E-2**	4.58E-2	4.55E-2
	DTLZ3	3.13E-1	**4.03E-2**	4.71E-2	4.66E0
	DTLZ4	3.85E-2	**3.71E-2**	4.87E-2	7.04E-2
	WFG6	**3.77E-2**	3.87E-2	4.63E-2	4.97E-2
	WFG7	**5.86E-2**	6.57E-2	6.79E-2	7.62E-2
15	DTLZ1	5.22E-2	**5.17E-2**	5.41E-2	1.04E-1
	DTLZ2	8.53E-2	**8.34E-2**	8.72E-2	9.82E-2
	DTLZ3	5.35E-1	**8.32E-2**	8.92E-2	4.56E0
	DTLZ4	7.77E-2	**7.38E-2**	8.92E-2	1.07E-1
	WFG6	1.36E-1	1.46E-1	1.27E-1	**1.19E-1**
	WFG7	**1.18E-1**	1.24E-1	1.2E-1	2.09E-1

For the Hypervolume (Table 5.4), the Synchronous version (SyncHeDi) was the best or equivalent to the best in 29 of 30 problems. In 21 instances, it was statistically better than the MOEAs applied alone. Besides, it was better, with statistical significance, than the Asynchronous version in 15 problems. The Asynchronous version (AsyncHeDi) was the best or equivalent in 14 problems, 9 of them with statistical significance to the MOEAs applied alone. The MOEA/D-STM executed independently was the best or equivalent in 9 problems, one of them better than the SyncHeDi and four of them better than the AsyncHeDi. The results obtained of hypervolume support the conclusions of the analysis of the IGD.

The Synchronous Heterogeneous Distributed approach for the cooperation of MOEAs (SyncHeDi) is better or equivalent than the best performing MOEA in almost all problems.

Table 5.4: The mean of Hypervolume. The best results are in boldface and a statistical equivalence to the best in a gray background.

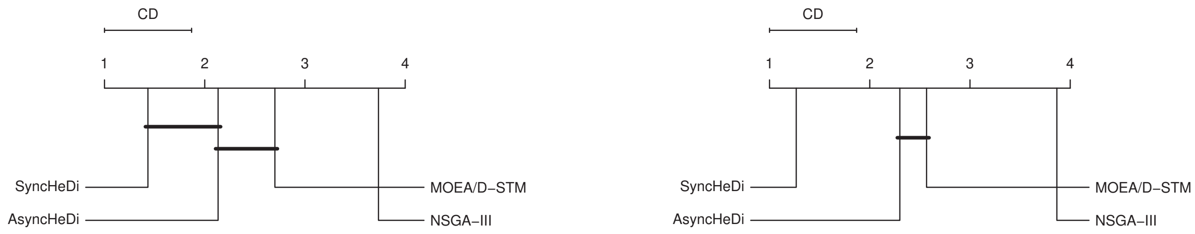
		HeDi		MOEAs applied alone	
Obj.	problem	Async	Sync	MOEA/D-STM	NSGA-III
3	DTLZ1	9.68E-1	**9.7E-1**	9.67E-1	9.16E-1
	DTLZ2	7.37E0	**7.38E0**	7.37E0	6.68E0
	DTLZ3	7.36E0	**7.38E0**	7.35E0	6.2E0
	DTLZ4	7.33E0	7.33E0	**7.36E0**	6.82E0
	WFG6	**6.77E1**	6.75E1	6.71E1	5.35E1
	WFG7	7.13E1	**7.21E1**	7.12E1	4.39E1
5	DTLZ1	9.96E-1	**9.98E-1**	9.54E-1	9.57E-1
	DTLZ2	3.16E1	**3.16E1**	3.15E1	2.96E1
	DTLZ3	1.04E1	**3.16E1**	1.97E1	2.49E1
	DTLZ4	3.16E1	**3.16E1**	3.15E1	3.07E1
	WFG6	7.59E3	**7.81E3**	7.18E3	4.17E3
	WFG7	7.82E3	**8.6E3**	7.66E3	4.36E3
8	DTLZ1	9.88E-1	9.95E-1	**9.95E-1**	6.26E-1
	DTLZ2	2.4E2	**2.41E2**	2.37E2	2.02E2
	DTLZ3	7.95E1	**2.4E2**	2.35E2	0E0
	DTLZ4	2.51E2	**2.53E2**	2.42E2	2.2E2
	WFG6	**2.3E7**	2.24E7	2.21E7	5.65E6
	WFG7	2.05E7	**2.32E7**	2.11E7	6.05E6
10	DTLZ1	9.95E-1	9.97E-1	**9.97E-1**	6.53E-1
	DTLZ2	**9.65E2**	9.62E2	9.39E2	8.75E2
	DTLZ3	8.44E2	**9.57E2**	9.27E2	0E0
	DTLZ4	1.01E3	**1.02E3**	9.49E2	8.22E2
	WFG6	9.29E9	**9.43E9**	8.78E9	2.1E9
	WFG7	8.51E9	**9.91E9**	8.79E9	2.72E9
15	DTLZ1	9.75E-1	9.78E-1	**9.85E-1**	6.15E-1
	DTLZ2	**2.62E4**	2.54E4	2.47E4	2.35E4
	DTLZ3	1.22E4	**2.56E4**	2.42E4	0E0
	DTLZ4	3.09E4	**3.13E4**	2.57E4	2.68E4
	WFG6	8.79E16	**9.56E16**	9.03E16	1.96E16
	WFG7	7.13E16	**8.11E16**	6.52E16	2.27E16

Moreover, the Asynchronous version was only worse than the best performing MOEA applied alone in 4 problem instances. Comparing the SyncHeDi to the AsyncHeDi, the Synchronous version performed better in both IGD and Hypervolume.

For better visualization of the results, we present the critical difference plot for both IGD and Hypervolume. Figure 5.1 (a) presents the critical difference plot for IGD. The Synchronous Heterogeneous Distributed approach for cooperation between MOEA/D-STM and NSGA-III (SyncHeDi) achieved the best average ranking. The SyncHeDi achieved an average ranking better than the MOEAs applied alone, with statistical significance. When compared to the Asynchronous version, the ranking was better but statistically equivalent. The Asynchronous version (AsyncHeDi) achieved an average ranking better than the MOEAs applied alone, but with statistical equivalence to the MOEA/D-STM. Then the MOEA/D-STM has a better ranking than the NSGA-III.

Figure 5.1 (b) presents the critical difference plot for the Hypervolume results. It is possible to observe that the order of the algorithms remained the same. However, the Synchronous version was better than the Asynchronous version with statistical significance. The Asynchronous version was again equivalent to the MOEA/D-STM. However, the difference is smaller for hypervolume than it was for IGD. Moreover, the same as for IGD, the MOEA/D-STM achieved a better average ranking than NSGA-III.

According to the presented results, it is possible to point out two main findings. First, the results indicate that the cooperation between MOEAs using the proposed Heterogeneous



(a) IGD

(b) Hypervolume

Figure 5.1: Critical difference plot for IGD and Hypervolume indicator. The algorithms connected by a bold line are statistically equivalent with 95% significance.

Distributed approach performs better than the MOEAs applied alone. This result is achieved due to the exchange of information between different strategies, improving the exploration of the search space, and avoiding local optima. Second, the synchronous communication performed better than the asynchronous version.

To understand this result, we investigated the execution of the algorithms. We observed that the NSGA-III island was about four to five times faster than the MOEA/D-STM on the asynchronous version. The effect of this difference in execution time was that, in the first stage, the NSGA-III executed about five iterations, sending solutions to MOEA/D-STM without receiving any. Until the end of the NSGA-III execution, it sends about four times more solutions than the received. The NSGA-III execution ends, while MOEA/D-STM has executed only about $1/4$ of its fitness evaluations. After that, the MOEA/D-STM runs the remaining $3/4$ of fitness evaluations.

In summary, the NSGA-III received only about $1/4$ of the information generated by MOEA/D-STM in this scenario. On the other hand, the MOEA/D-STM gets, at the beginning of the search, all the knowledge produced by NSGA-III. Then, it executes most of the time without receiving any more external information. Therefore, when there is no synchronization of the algorithms, the exchange of information is affected, mainly when there is a significant difference in the execution time of the algorithms.

5.2.2 Discussion

In this section, we evaluate a Heterogeneous Distributed Framework for the collaboration of Many-Objective Evolutionary Algorithms. The distribution is implemented using the Heterogeneous Island-model, where each island executes a different MOEA. Two communication strategies were implemented: synchronous and asynchronous. The framework was evaluated using two state-of-the-art MOEAs and six benchmark problems, varying the number of objectives from 3 to 15. The quality indicators, IGD and Hypervolume, were used to evaluate the results achieved by the MOEAs.

The cooperation between the algorithms could improve the convergence and diversity of the algorithms in most problem instances. Observing both quality indicators, hypervolume, and IGD, it is possible to conclude that the framework with synchronous communication has better results than the best MOEA applied alone with statistical significance. When comparing the synchronous and asynchronous communication, the synchronous showed better results. The significant difference between the computational cost of the chosen MOEAs led to this behavior. This execution time difference directly influenced the quality of the information exchange.

In general, we conclude that the Heterogeneous Distributed framework is a simple and effective manner to allow different MOEAs to solve a complex problem. For further experiments, we evaluate the use of hyper-heuristics. We make use of several MOEAs. Although

MOEA/D-STM was not used due to its computational cost. The NSGA-III implementation was replaced by one providing better results. The use of hyper-heuristics solves some disabilities of HeDi. First, we had difficulties with the scalability of HeDi, since as higher the number of MOEAs smaller the population sizes. Also, it was difficult to include other types of MOEAs, since the use of weight vectors was required to guide the different populations during the search. See Appendix A and B for more details about the difficulties observed for HeDi, the comparison of different implementations of NSGA-III, and preliminary experiments to validate the use of hyper-heuristics.

5.3 PRELIMINARY ANALYSIS WITH HYPER-HEURISTICS

This section presents the experimental analysis used to evaluate the information exchange in the context of hyper-heuristics. For this analysis, we built two hyper-heuristic frameworks: HHcMOEA, which includes the migration step, and HHMOEA, without the information exchange (both presented at Section 4.4). When the migration step is not applied, the updated population is sent to the next MOEA only. Further, the pool of low-level heuristics includes seven MOEAs: SPEA2 (Zitzler et al., 2001), NSGA-II (Deb et al., 2002), MOEA/D (Zhang and Li, 2007), NSGA-III (Deb and Jain, 2014), MOEA/DD (Li et al., 2015c), MOMBI2 (Gómez and Coello, 2015), and ThetaDEA (Yuan et al., 2016)[1].

This experiment uses a set of six minimization benchmark problems (and its maximization version). Those problems have a diversity of characteristics and were picked from Ishibuchi et al. (2017). Further, this set avoids problem instances too similar in the Pareto front shape, for example, WFG4 to WFG9. The properties of the benchmark problems used in this analysis are presented in Section 2.1.7. We set the number of objectives to $m = \{3, 5, 8, 10\}$. The number of iterations varies by problem instance and the number of objectives, and it is presented in Table 5.5. The population size was set according to the approach presented in Section 2.1.8. However, only one population is executed for each iteration. Thus, the number of total fitness evaluations is the same used for one MOEA applied independently. The number of iterations and population size for conventional problems is used for inverted problems. Finally, the number of independent runs was set to 20. This experimental setup is based on (Deb and Jain, 2014; Li et al., 2015c; Ishibuchi et al., 2017). On the proposed approach, the only parameters are from the roulette heuristic selection. It was configured as a minimum probability of 0.002 and probability increment of p/inc, where p is 1 divided by the number of MOEAs in the pool, and inc is a fixed parameter of 5.0.

Table 5.5: Number of iterations for different problem instances.

	Number of objectives (m)			
Problem	3	5	8	10
DTLZ1	400	600	750	1000
DTLZ2	250	350	500	750
WFG1-4	400	750	1500	2000

The results of the algorithms are evaluated using the hypervolume quality indicator (see Section 2.1.6). The reference point used in this paper is based on the literature of the benchmark

[1]In this moment of the research, there was no implementation of SPEA2SDE and HypE available for the framework used (jMetal), see Section 4.2

problems (Li et al., 2015c; Ishibuchi et al., 2017): for DTLZ1 we used the point $(1.0, \ldots, 1.0)$; for DTLZ2 we used $(2.0, \ldots, 2.0)$; for WFG problems we used $(3.0, \ldots, 2.0 \times m + 1.0)$; for the inverted problems we used $(0.0, \ldots, 0.0)$. The following section presents the average hypervolume tables and the critical difference plot to compare the hyper-heuristic with and without the information exchange step.

5.3.1 Preliminary analysis results

In this section, Table 5.6 presents the average hypervolume value for each problem instance. The first important note is that each MOEA achieved the best average hypervolume in at least one problem instance. This observation demonstrates the diversity of characteristics of the benchmark set. The results achieved by the MOEAs applied independently are coherent to the ones obtained by (Ishibuchi et al., 2017). NSGA-III was the one that performed the best balance between conventional and inverted problems. It was the best or equivalent in $1/3$ of instances for conventional benchmark instances and $1/3$ for the inverted cases. On the other hand, ThetaDEA was biased to the conventional problems (best or equivalent in 71% of traditional benchmark instances). It was the best or equivalent in only 21% of the problem instances on the inverted problems.

Similarly to NSGA-III, NSGA-II also demonstrated a good balance between traditional and inverted problem instances. It shows that this MOEA is not always a bad option for many-objective optimization. MOMBI2 and MOEA/DD were also biased toward conventional problems. For example, MOEA/DD was the best or equivalent in 14 of 24 instances on the original problems. Still, on the inverted problems, it was always statistically worse than the best performing MOEA. On the overall analysis, MOEA/D and SPEA2 were the ones with the worst results. However, considering only the inverted problems, they achieved a performance better or similar to state-of-art MOEAs such as ThetaDEA, MOMBI2, and MOEA/DD.

Those observations exemplify the presence of the NFL theorem on many-objective optimization. Besides, it demonstrates the bias between some state-of-art MOEAs and the traditionally used benchmark problems. Finally, dominance-based MOEAs may achieve better results than the state-of-art MOEAs, depending on the problem instance. Those conclusions support the study of hyper-heuristics and the exchanging of information for many-objective optimization. This kind of research may improve the generality of state-of-art MOEAs and enhance the quality of the results.

Next, we compared the MOEAs applied alone and the two versions of the preliminary hyper-heuristic model. HHcMOEA includes a migration phase. HHMOEA does not use migration (the updated population is sent to the next MOEA only). HHcMOEA achieved the best average hypervolume value for 70.83% of instances. It was also the best, or equivalent to the best in 95.83% of problem instances. HHcMOEA achieved worse results for WFG1 with 5 and 8 objectives (with a significant difference) than the best MOEA (MOMBI2). On the other hand, HHMOEA (no migration step) achieved the best average hypervolume in only one instance (2.08%), being the best or equivalent to the best in 62.50% of problem instances. Fig 5.2 presents the critical difference plot for the hypervolume indicator. The best average ranking was achieved by the hyper-heuristic using migration (HHcMOEA). It was, on the overall evaluation, better than all other algorithms with statistical significance. The hyper-heuristic model without migration (HHMOEA) achieved the second-best average ranking but statistically equivalent to NSGA-III, ThetaDEA, NSGA-II, and MOMBI2. The algorithms MOEA/DD, MOEA/D, and SPEA2, achieved the worst results.

Table 5.6: Average values for hypervolume: two version of collaboration guided by hyper-heuristics (HHcMOEA and HHMOEA) and the seven MOEAs that compose the pool of heuristics. The best value of each problem instance is highlighted in bold face; gray background means statistically equivalent to the best with 95% significance.

Obj.	problem	HHMOEA	HHcMOEA	MOMBI2	ThetaDEA	NSGA-III	MOEA/DD	NSGA-II	SPEA2	MOEA/D
3	DTLZ1	0.971584	**0.974717**	0.967816	0.973144	0.973301	0.973513	0.968926	0.966631	0.966107
	DTLZ2	7.41218	**7.42363**	7.39182	7.41279	7.41254	7.41330	7.35247	7.30016	7.36957
	WFG1	50.6037	56.5215	**60.7434**	57.4383	54.2592	56.4830	53.2943	43.8565	42.5854
	WFG2	94.7253	**99.6281**	89.6473	96.2105	90.0472	90.7819	91.3203	93.7578	93.4973
	WFG3	72.2592	**74.0992**	72.2889	72.7250	71.9752	69.8021	73.6184	70.3799	68.0672
	WFG4	73.5246	74.8627	74.6585	**74.9468**	74.8198	74.7260	72.0319	66.8436	66.0525
	MinusDTLZ1	1.85481e+07	**2.01372e+07**	1.59594e+07	1.57550e+07	1.70084e+07	1.57276e+07	1.89605e+07	2.00680e+07	1.23646e+07
	MinusDTLZ2	19.2508	**19.8085**	18.8133	18.9825	18.5487	18.7867	18.5069	18.7815	18.3230
	MinusWFG1	25.9686	**28.5620**	8.13247	11.5160	14.8120	6.53517	17.8856	10.0736	27.3187
	MinusWFG2	53.5835	**54.7252**	54.0058	54.2551	54.2088	54.0335	52.7348	51.3901	51.8712
	MinusWFG3	36.1199	**38.9133**	37.3967	35.7614	35.9344	33.1513	35.7741	34.4826	32.4507
	MinusWFG4	64.1576	**68.2980**	66.4773	65.1204	62.4368	62.5788	63.6073	61.9582	64.9987
5	DTLZ1	0.979010	0.998933	0.996571	0.998969	0.998960	**0.998977**	0.00000	0.00000	0.906870
	DTLZ2	31.6922	**31.7041**	31.6555	31.6961	31.6927	31.6969	30.4208	31.1851	31.5208
	WFG1	6440.35	6135.08	**7936.69**	7268.16	5929.32	6763.78	5041.39	3829.07	5600.27
	WFG2	10055.1	**10328.3**	9729.91	9976.01	10064.6	10074.6	10269.1	9875.54	9917.50
	WFG3	7072.45	**7245.49**	6939.74	6955.41	6798.10	6453.06	7166.60	5274.62	6101.73
	WFG4	8672.12	**8962.03**	8922.26	8878.68	8818.41	8901.93	7557.82	6766.02	6591.53
	MinusDTLZ1	4.61160e+10	4.53569e+10	8.41459e+08	4.33262e+07	9.38298e+09	6.42408e+09	4.05590e+10	**5.82908e+10**	6.58647e+09
	MinusDTLZ2	26.5023	**31.8601**	7.26639	19.5276	15.7131	4.63808	29.6240	26.1070	21.3842
	MinusWFG1	104.666	**149.935**	40.9386	55.0185	64.7106	10.0331	67.7377	41.8101	121.793
	MinusWFG2	343.385	**416.241**	303.645	304.878	351.208	263.141	355.043	327.713	297.996
	MinusWFG3	273.047	**344.357**	222.997	189.521	269.945	147.512	287.937	271.598	203.219
	MinusWFG4	1573.81	**1983.84**	1211.30	1566.85	1536.94	1200.90	1519.79	1011.16	1296.40
8	DTLZ1	0.999191	0.999936	0.995889	0.999969	**0.999974**	0.999947	0.00000	0.00000	0.995011
	DTLZ2	255.793	**255.843**	255.482	255.832	255.824	255.830	21.0374	25.6580	237.230
	WFG1	2.19585e+07	2.57764e+07	**3.02974e+07**	2.98614e+07	2.71509e+07	2.50014e+07	1.50820e+07	1.05066e+07	2.57562e+07
	WFG2	3.27711e+07	3.42370e+07	3.06318e+07	3.03619e+07	3.15724e+07	3.24991e+07	**3.42511e+07**	3.06606e+07	3.32591e+07
	WFG3	2.20728e+07	**2.34516e+07**	3.68399e+06	1.68018e+07	2.05292e+07	1.85675e+07	2.27804e+07	1.13171e+07	1.95878e+07
	WFG4	2.84114e+07	3.04679e+07	2.92520e+07	3.13874e+07	**3.13976e+07**	3.04079e+07	2.02148e+07	1.63987e+07	1.87389e+07
	MinusDTLZ1	**8.01995e+14**	5.54905e+14	1.54544e+14	1.15168e+14	3.81465e+14	1.66232e+12	4.85612e+13	7.65409e+13	1.17105e+12
	MinusDTLZ2	8.69592	**10.6049**	4.20588	3.47354	5.24108	0.826540	6.09313	2.03880	0.0263792
	MinusWFG1	358.603	**646.142**	16.9479	124.036	129.186	18.9656	165.767	129.386	470.652
	MinusWFG2	1470.33	**1985.68**	815.593	1277.28	1417.81	361.871	1516.18	1401.69	986.383
	MinusWFG3	2669.86	**3658.24**	1743.08	1380.16	2951.94	539.684	1694.66	3072.07	1128.33
	MinusWFG4	150823	**196685**	110908	168576	151942	21970.8	104285	30353.3	25662.0
10	DTLZ1	0.999351	0.999997	0.995530	0.999996	**0.999998**	0.999993	0.00000	0.00000	0.996563
	DTLZ2	1023.24	1023.91	1022.30	1023.92	1023.91	**1023.92**	53.3866	77.4728	938.637
	WFG1	1.09519e+10	1.28954e+10	1.26580e+10	1.23742e+10	1.18121e+10	1.14905e+10	7.08265e+09	3.91531e+09	**1.30579e+10**
	WFG2	1.33153e+10	**1.37181e+10**	1.24930e+10	1.23371e+10	1.26236e+10	1.31043e+10	1.37075e+10	1.21469e+10	1.36884e+10
	WFG3	9.24243e+09	**9.71723e+09**	5.65469e+09	8.17914e+09	7.52058e+09	7.11934e+09	9.20566e+09	4.10326e+09	8.61037e+09
	WFG4	1.17718e+10	1.26096e+10	1.18402e+10	**1.30054e+10**	1.29986e+10	1.22769e+10	7.34597e+09	6.85962e+09	8.58168e+09
	MinusDTLZ1	3.55871e+17	**4.42036e+17**	8.03618e+16	1.35685e+16	3.82748e+17	1.26631e+14	3.11683e+15	7.33207e+15	1.19174e+14
	MinusDTLZ2	4.76030	**7.06653**	2.80922	4.74477	2.76230	0.438842	1.91613	0.273260	0.000264356
	MinusWFG1	853.997	**1543.94**	16.9758	237.332	223.507	32.3711	366.558	399.555	1057.56
	MinusWFG2	3441.17	**5306.24**	1248.93	2932.28	3035.74	405.590	3710.25	3699.94	2119.50
	MinusWFG3	13659.6	**26555.1**	7804.64	6940.34	18332.2	1179.26	7167.40	19308.7	4249.06
	MinusWFG4	4.27927e+06	**6.13205e+06**	3.08929e+06	3.30907e+06	5.84987e+06	167831	2.09628e+06	408098	195203

5.3.2 Discussion

This section presented a preliminary analysis for the collaboration of MOEAs for many-objective optimization problems. Two hyper-heuristic frameworks were evaluated, HHcMOEA and HHMOEA. Here, the goal is to prioritize a MOEA that shows a better quality of solutions, ensuring a probability to reevaluate the others. This quality was assessed using a fitness improvement rate metric based on the R2 improvement rate. One interesting observation is that every MOEA executed independently achieved the best results in at least one problem instance. This demonstrates the diversity of MOEAs applied, also the variety of characteristics of the set of benchmark problems. Compared to the MOEAs executed independently, hyper-heuristics achieved the best, or equivalent to the best results, in almost all problem instances. Also, compared the two versions, with and without the exchange of information step. The results were favorable to the version with the exchange of information. On the general evaluation, the proposed model of collaboration achieved the best overall result with statistical significance compared to the MOEAs executed alone and to the version without migration.

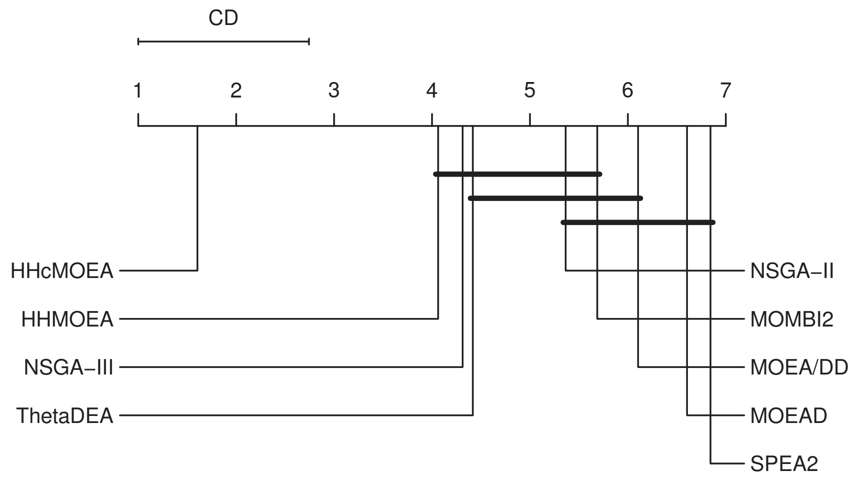


Figure 5.2: Critical difference plot of the HV indicator. Algorithms connected by a bold horizontal line are considered statistically equivalent with 95% significance.

Therefore, it is possible to conclude that HHcMOEA achieved better results than each one of the evaluated MOEAs applied independently. Furthermore, we conclude that this preliminary hyper-heuristic framework achieved better results when using the migration step. These results indicate that exchanging information can keep the MOEA updated, even if it has not been applied for a while. The disadvantage of the proposed approach is the increase in the computational cost due to the migration step. Although a parallel implementation of this procedure may alleviate this issue.

5.4 EMPIRICAL ANALYSIS OF THE IMPROVED APPROACH: HH-CO

Motivated by the results presented in Section 5.3 and other experiments throughout this research, we improved the HHcMOEA and proposed the HH-CO. This section describes the evaluation of the proposed HH-CO compared to a state-of-the-art hyper-heuristic for multi-objective optimization, the HH-LA. To the extent of our knowledge, HH-LA is one of the best performing hyper-heuristic in the literature for selecting MOEAs. It was favorable compared to HH-CF (choice function) and HH-RC (random choice) (Li et al., 2018b). Also, we compare the hyper-heuristics to the MOEAs from the pool. In this analysis, we used the MaF benchmark set (see Section 2.1.7). Finally, we compare HH-CO to the winners of the CEC'18 competition: CVEA3, BCE-IBEA, and AMPDEA[2]. On the many-objective competition, and so in this study, two quality indicators are used to evaluate the algorithms: IGD and hypervolume[3] (see Section 2.1.6).

Furthermore, in the MaF methodology, the output size must be truncated to the limit of 240 solutions (Cheng et al., 2018). The output of HH-CO is a combination of the population of each MOEA in the pool, removing the dominated and the repeated, truncated to a subset of evenly distributed solutions. Moreover, the output size of HH-CO algorithms has the same size as the other algorithms[4]. For example, for fifteen objectives, the output size is 135 for comparisons

[2]The Pareto fronts from CVEA3, BCE-IBEA, and AMPDEA were downloaded from the competition repository: `https://github.com/ranchengcn/IEEE-CEC-MaOO-Competition`

[3]The reference Pareto front used to compute IGD is available on the competition website: `https://www.cs.bham.ac.uk/~chengr/CEC_Comp_on_MaOO/2018/webpage.html`

[4]However, it was observed that the output without truncation leads to better results.

to the pool and 240 for comparisons to the competition winners. Finally, the population size was configured following the settings from Section 2.1.8.

The parameter configuration of HH-LA follows the guidelines from the paper where it was proposed (Li et al., 2018b). The exploration phase $\tau = 0.5$, the reward/penalty multiplier $m = 2.5$, the maximum number of iterations for applying a low-level MOEA $K = 3$, and the quality indicator improvement threshold $\Delta_v = 0.0075$ (Li et al., 2018b). On the other hand, the proposed HH-CO has no parameters to be set.

5.4.1 Differences and similarities of HH-CO to HH-LA

The main difference between HH-CO and HH-LA is: in HH-LA, the selected MOEA is initialized with an external population and executed during K iterations. Then, the MOEA outputs the population back. This population is then given as the initial population for the next selected MOEA. On the other hand, in HH-CO, the selected MOEA executes its internal population during one iteration and shares the newly generated solutions to every other MOEA. Besides, HH-CO applies the MOEA with the most significant improvement in the last iteration. Differently, in HH-LA, the selection is based on a Learning Automata (LA). Some parameters were set to the same values to compare the HH-CO to the HH-LA framework. In this work, HH-LA uses the same pool of nine MOEAs as HH-CO. The computation of reward is given by the R2 improvement, as in HH-CO. Therefore, we evaluate the differences between the procedures of the frameworks.

5.4.2 Comparing HH-CO, HH-LA, and the low-level heuristics

First, we compared the HH-CO to a state-of-the-art hyper-heuristic (HH-LA) and the nine MOEAs that compose the pool of heuristics. In Table 5.7, the average values of the IGD indicator are presented. For the IGD indicator, the proposed hyper-heuristic achieved the best or equivalent to the best result in 43 out of 45 problem instances (95.56% of the cases). Mainly for 15 objectives, HH-CO achieved the best result in 7 and equivalent to the best in 8 out of 15 problems. There were only two instances where HH-CO was not the best or equivalent to the best. One of them is the MaF14 with five objectives, a large-scale problem with a complicated fitness landscape and mixed variable separability. In this instance, the best result was achieved by MOEA/DD, equivalent to SPEA2SDE, HypE, and the HH-LA. The other instance was MaF05, with ten objectives. MaF05 has a highly biased distribution and a badly-scaled Pareto front. In this instance, the best result was achieved by the HH-LA, equivalent to NSGA-III, ThetaDEA, NSGA-II, and SPEA2SDE. The state-of-the-art hyper-heuristic, HH-LA, produced the best IGD average in just one problem instance and equivalent to the best in other 9 out of 45. HH-LA presented promising results for the IGD indicator, mainly in MaF01 (inverted Pareto front) and MaF05.

On the other hand, the proposed HH-CO achieved good results mainly in MaF15 (inverted Pareto front and complicated fitness landscape) and MaF10 (complicated mixed geometries). When compared to the MOEAs that compose the pool, it was possible to observe that most of them achieved the best IGD average value in at least one problem instance. These results demonstrate the diversity of the MOEAs that compose the pool and the variability of the benchmark set characteristics.

In Figure 5.3, the Critical Difference plot for the IGD indicator is presented. According to this analysis, the proposed HH-CO presented the best average ranking for IGD, statistically equivalent to SPEA2SDE. On the other hand, the state-of-the-art HH-LA was statistically worse than the best MOEA and equivalent to seven others. SPEA2SDE was the best performing MOEA, equivalent to NSGA-II and NSGA-III. Besides, NSGA-II, known as having poor performance on

Table 5.7: Average values for IGD comparing HH-CO, HH-LA and nine MOEAs

Obj.	problem	HH-CO	HH-LA	SPEA2-SDE	HypE	MOMBI2	MOEA/DD	ThetaDEA	NSGA-III	MOEA/D	SPEA2	NSGA-II
	MaF01	1.32E-3	1.34E-3	**1.16E-3**	2.16E-3	2.43E-3	2.34E-3	2.38E-3	1.95E-3	1.94E-3	1.34E-3	1.6E-3
	MaF02	7.12E-3	1.04E-2	7.12E-3	1.38E-2	1.06E-2	7.32E-3	9.6E-3	9.09E-3	1.1E-2	**6.79E-3**	9.79E-3
	MaF03	1.3E-2	1.07E-2	1.24E-3	5.73E-2	4.24E-3	1.6E-3	1.41E-3	**8.35E-4**	1.01E-2	3.73E8	5.75E2
	MaF04	2.13E-2	8.21E-2	3.81E-2	3.18E-2	3.27E-2	6.75E-1	3.32E-2	3.62E-2	5.03E-2	**1.88E-2**	2.12E-2
	MaF05	2.32E-2	2.7E-2	2.8E-2	8.62E-2	**2.25E-2**	5.93E-2	2.31E-2	2.31E-2	6.93E-2	2.43E-2	2.52E-2
	MaF06	5.34E-5	5.12E-4	9.65E-5	2.55E-3	3.71E-3	9.15E-4	1.25E-3	5.94E-4	4.32E-4	**2.7E-5**	4.03E-5
	MaF07	**2.78E-3**	4.68E-3	3.13E-3	5.73E-3	4.13E-3	6.04E-3	3.48E-3	3.42E-3	7.3E-3	3.16E-3	3.33E-3
5	MaF08	1.18E-3	2.73E-3	1.3E-3	9.91E-3	4E-3	2.04E0	4.59E-3	3.18E-3	1.44E-3	**1.14E-3**	1.71E-3
	MaF09	1.36E-3	1.31E-3	**1.05E-3**	7.84E-3	4.58E-3	1.11E-2	1.06E-2	6.79E-3	1.63E-3	1.22E-3	7.17E-3
	MaF10	**7.05E-3**	1.58E-2	1.12E-2	2.16E-2	7.85E-3	1.38E-2	9.17E-3	1.07E-2	2.46E-2	1.62E-2	8.4E-3
	MaF11	8.16E-3	1.83E-2	2.16E-2	9.2E-2	1.65E-2	5.99E-2	1.32E-2	1.1E-2	1.77E-2	**7.5E-3**	9.58E-3
	MaF12	1.14E-2	1.32E-2	1.23E-2	4.86E-2	1.18E-2	1.17E-2	**1.06E-2**	1.06E-2	3.25E-2	1.13E-2	1.25E-2
	MaF13	**9.24E-4**	2.53E-3	1.12E-3	3.97E-3	6.66E-3	1.93E-3	3.6E-3	3.88E-3	1.48E-3	1.23E3	1.68E-3
	MaF14	6.4E-3	5.94E-3	4.74E-3	5.78E-3	6.96E-3	**4.1E-3**	1.44E-2	1.47E-2	7.85E-3	5.3E1	2.74E-1
	MaF15	**2.99E-3**	8.42E-3	3.66E	6.71E-3	4.28E-3	5.13E-3	1.14E-2	1.26E-2	3.25E-3	8.9E-2	2.95E-1
	MaF01	2.89E-3	3.41E-3	**2.7E-3**	4.25E-3	4.91E-3	6.13E-3	4.15E-3	4.03E-3	3.55E-3	3.45E-3	3.52E-3
	MaF02	2.08E-3	4.27E-3	2.15E-3	4.78E-3	6.23E-3	3.42E-3	2.67E-3	2.66E-3	3.75E-3	2.09E-3	**2.04E-3**
	MaF03	1.09E-1	4.11E5	**1.74E-3**	3.82E1	6.17E-3	1.84E-3	2.03E-3	5.69E-3	2.18E-3	1.7E10	4.23E3
	MaF04	**7.72E-1**	4.07E0	1.97E0	1.03E0	1.95E0	9.84E1	1.71E0	1.87E0	3.6E0	7.72E-1	8.02E-1
	MaF05	1.82E0	**1.15E0**	1.7E0	2.43E0	4.29E0	4.39E0	1.23E0	1.18E0	4.45E0	2.08E0	1.27E0
	MaF06	**1.98E-4**	1.95E-2	5.63E-3	2.51E-3	6.53E-3	2.11E-3	2.86E-3	3.94E-3	2.84E-4	1.46E0	3.99E-3
	MaF07	**7.27E-3**	1.44E-2	7.49E-3	5.65E-2	1.51E-2	1.48E-2	8.27E-3	1.2E-2	9.17E-3	1.77E-2	1.19E-2
10	MaF08	1.61E-3	3.66E-3	1.72E-3	5.88E-3	1.46E-2	2.27E-2	1.1E-2	5.4E-3	1.9E-3	**1.52E-3**	2.1E-3
	MaF09	3.03E-3	1.02E-2	**1.38E-3**	2.12E	1.43E-2	6.87E-2	1.11E-2	1.2E-2	3.43E	6.07E-1	4.4E-1
	MaF10	**1.83E-2**	4.31E-2	2.48E-2	4.26E-2	2.49E-2	3.44E-2	2.16E-2	2.41E-2	3.93E-2	3.64E-2	2.19E-2
	MaF11	2.51E-2	1.3E-1	6.73E-2	1.29E	7.04E-2	1.14E-1	3.32E-2	3.71E-2	1.01E-1	**1.56E-2**	1.6E-2
	MaF12	5.48E-2	6.27E-2	**5.46E-2**	9.05E-2	6.53E-2	8.15E-2	5.68E-2	5.73E-2	8.55E-2	5.85E-2	6.12E-2
	MaF13	1.44E-3	3.19E-3	1.2E-3	3.97E-3	7.95E-3	4.35E-3	6.69E-3	4.91E-3	3.17E-3	**1.16E-3**	1.75E-3
	MaF14	8.34E-3	4.45E-1	**4.47E-3**	1.46E0	9.55E-3	8.95E-3	4.5E-2	4.54E-2	9.45E-3	3.61E2	1.18E0
	MaF15	**8.97E-3**	1.28E-2	9.22E-3	1.13E	1.27E-2	1.25E-2	1.54E-2	1.05E-2	5.57E-2	3.45E0	1.29E0
	MaF01	**4.31E-3**	4.75E-3	4.49E-3	7.32E-3	5.99E-3	7.65E-3	4.93E-3	5.03E-3	5.43E-3	5.19E-3	4.89E-3
	MaF02	2.54E-3	6.44E-3	3.3E-3	7.27E-3	1.07E-2	4.65E-3	4.15E-3	3.92E-3	5.2E-3	3.32E-3	**2.51E-3**
	MaF03	2.63E-2	3.09E7	2.12E-3	9.4E5	6.94E-3	**2.06E-3**	4.57E-3	7.41E-3	2.45E-3	2.28E10	4.21E2
	MaF04	**2.66E1**	3.47E2	9.58E1	9.98E2	1.04E2	3.2E3	8.34E1	6.91E1	5.07E1	2.88E1	3.36E1
	MaF05	4.48E1	5.75E1	5.07E1	9.86E1	1.25E2	1.17E2	5.5E1	5.5E1	1.25E2	7.67E1	**3.6E1**
	MaF06	3.71E-4	2.64E-1	4.4E-3	2.42E-1	7.91E	1.98E-3	3.56E-3	4.48E-3	**3.35E-4**	2.03E0	4.62E-3
	MaF07	1.32E-2	2.36E-2	**1.23E-2**	3.68E-1	5.2E-2	2.6E-2	5.64E-2	6.84E-2	1.52E-2	7.35E-2	2.62E-2
15	MaF08	**2.32E-3**	1.17E0	3.11E-3	8.25E-3	2.8E-2	7.9E0	1.54E-2	7.43E-3	2.44E-3	2.43E-3	3.56E-3
	MaF09	3.44E-3	7.39E-3	**2.25E-3**	6.83E-2	6.78E-2	8.01E-2	4.04E-2	1.71E-2	1.02E-2	6.62E-3	7.8E-2
	MaF10	**2.9E-2**	5.84E-2	4.73E-2	5.78E-2	5.5E-2	5.06E-2	3.34E-2	3.38E-2	5.34E-2	5.46E-2	3.15E-2
	MaF11	2.74E-2	1.86E-1	1.55E-1	2.19E-1	2.01E-1	2.05E-1	1.38E-1	6.04E-2	2.05E-1	3.74E-2	**1.59E-2**
	MaF12	**1.18E-1**	1.52E-1	1.19E-1	2.26E-1	1.43E-1	1.5E-1	1.21E-1	1.21E-1	1.74E-1	1.32E-1	1.21E-1
	MaF13	**1.75E-3**	6.01E-3	1.79E-3	5.28E-3	1.1E-2	6.07E-3	8.01E-3	6.59E-3	5.63E-3	2.63E4	4.53E-3
	MaF14	1.1E-2	1.65E1	**5.59E-3**	2.23E-2	9.34E-3	7.54E-3	1.56E-2	1.9E-2	1.21E-2	6.62E2	2.97E-1
	MaF15	**1.26E-2**	4.12E-2	1.36E-2	1.4E-2	1.96E-2	1.53E-2	1.84E-2	6.72E-2	1.1E	4.39E0	8.52E-1

many-objective problems, achieved better results than state-of-the-art MOEAs, such as HypE, MOMBI2, and MOEA/DD. This result demonstrates that Pareto based MOEAs can achieve good results on many-objective optimization depending on the problem instance characteristics.

We also computed the hypervolume indicator (HV), calculated approximated using Monte Carlo sampling. The average HV per problem instance is presented in Table 5.8. The proposed HH-CO achieved the best or equivalent result in 80%, 36 out of 45 instances (for IGD, 95%). The HH-CO results were better for the number of objectives equal to 15 than 5 and 10. For the instances with 15 objectives, HH-CO achieved the best HV average in 9 and equivalent in other 5. The state-of-the-art hyper-heuristic HH-LA produced the best hypervolume average in just one problem instance (MaF01 with 5 objectives) and equivalent to the best result in another 9 out of 45 (20%). Besides, the best performing MOEA was SPEA2SDE. It achieved the best hypervolume average in 16 of 45 instances. When we observe the Critical Difference plot (Figure 5.4), the HH-CO achieved the best average ranking, statistically equivalent only to SPEA2SDE. On the other hand, HH-LA was equivalent to 8 MOEAs and worse than SPEA2SDE.

The overall observations from IGD and hypervolume are similar. First, in the evaluated benchmark, the HH-CO presented better results than HH-LA. These results demonstrate the

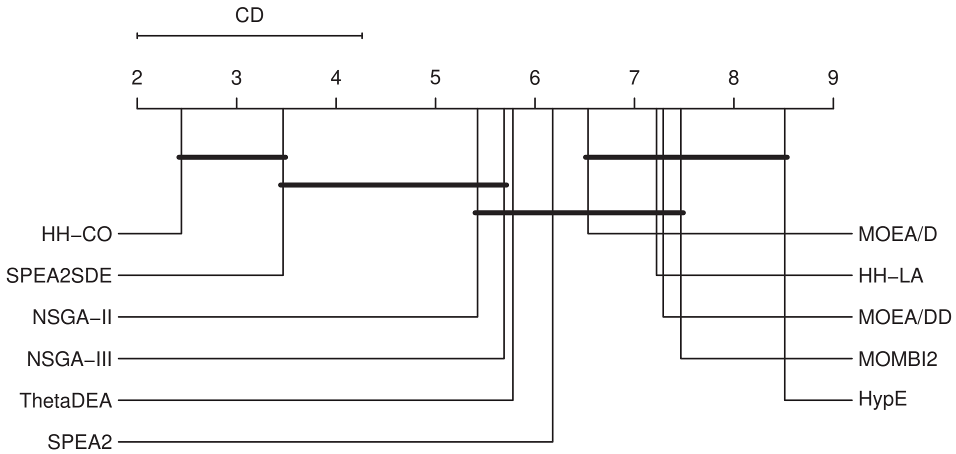


Figure 5.3: Critical Difference plot for IGD comparing HH-CO, HH-LA and nine MOEAs.

Table 5.8: Average values for hypervolume comparing HH-CO, HH-LA and nine MOEAs

Obj.	problem	HH-CO	HH-LA	SPEA2-SDE	HypE	MOMBI2	MOEA/DD	ThetaDEA	NSGA-III	MOEA/D	SPEA2	NSGA-II
	MaF01	9.31E-4	**1.7E-3**	1.48E-3	1.58E	7E-5	8.45E-8	2.47E-7	1.2E-4	1.98E-4	6.77E-4	4.68E-4
	MaF02	3.45E-1	3.12E-1	**3.88E-1**	2.97E-1	2.44E-1	3.4E-1	3.12E-1	3.24E-1	2.14E-1	3.06E-1	2.58E-1
	MaF03	8.42E-1	7.48E-1	9.9E-1	2.57E-1	7.32E-1	9.91E-1	9.88E-1	**9.99E-1**	9.4E-1	0E0	0E0
	MaF04	4.17E-2	4.19E-2	**6.17E-2**	2.85E-2	1.37E-2	0E0	2.79E-2	2.14E-2	5.06E-3	4.3E-2	4.8E-2
	MaF05	6.12E-1	6.54E-1	6.42E-1	3.42E-1	6.93E-1	5.34E-1	**6.98E-1**	6.98E-1	2.55E	5.52E-1	4.54E-1
	MaF06	3.47E-2	2.85E-2	3.41E-2	1.17E-2	1.57E-2	1.81E-2	2.27E-2	2.72E-2	2.5E-2	3.47E-2	**3.48E-2**
	MaF07	1.72E-1	7.16E-2	**2.31E-1**	8.08E-2	2.05E-1	5.75E-3	1.29E-1	1.98E-1	5.48E-3	1.14E-1	1.38E-1
5	MaF08	8.11E-2	6.77E-2	**8.46E-2**	3.79E-2	5.22E-2	2.27E-2	4.88E-2	6.05E-2	7.69E-2	8.18E-2	7.34E-2
	MaF09	2.33E-1	2.33E-1	**2.4E-1**	1.13E-1	1.4E-1	6.86E-2	7.83E-2	1.21E-1	2.26E-1	2.34E-1	1.13E-1
	MaF10	9.25E-1	8.84E-1	9.4E-1	**9.84E-1**	9.65E-1	9.53E-1	9.22E-1	9.12E-1	6.17E-1	3.85E-1	8.96E-1
	MaF11	**9.94E-1**	9.81E-1	9.81E-1	9.81E-1	9.91E-1	9.66E-1	9.93E-1	9.94E-1	9.74E-1	9.72E-1	9.83E-1
	MaF12	5.79E-1	5.87E-1	6.19E-1	2E-1	6.31E-1	5.95E-1	**6.46E-1**	6.39E-1	2.16E-1	4.75E-1	4.52E-1
	MaF13	2.37E-1	2.19E-1	**2.41E-1**	1.61E-1	7.4E	1.9E-1	1.02E-1	7.94E-2	2.22E-1	1.88E-1	1.85E-1
	MaF14	2.79E-1	4.98E-1	**6.8E-1**	4.7E-1	4.37E-1	5.24E-1	6.39E-2	2.1E-2	3.53E-2	0E0	0E0
	MaF15	1.09E-2	2.04E-2	**6.34E-2**	3.75E-4	2.09E-2	6.97E-3	9.51E-4	0E0	9.54E-3	0E0	0E0
	MaF01	2.16E-1	1.04E-1	2.13E-1	**2.22E-1**	6.16E-2	6.89E-2	6.01E-2	6.32E-2	2.2E-1	1.5E-1	2.06E-1
	MaF02	2.43E0	2.26E0	**2.46E0**	1.93E0	2.04E0	2.41E0	2.43E0	2.45E0	2.39E0	2.31E0	2.44E0
	MaF03	2.46E0	1.54E0	2.59E0	0E0	2.19E0	2.58E0	2.59E0	2.43E0	**2.59E0**	0E0	0E0
	MaF04	3.58E-1	2.55E-1	1.53E-1	2.71E-1	1.09E-1	0E0	2.07E-1	2.23E-1	1.74E-1	3.2E-1	**3.87E-1**
	MaF05	2.49E0	2.38E0	2.57E0	2.25E0	2.57E0	2.55E0	**2.59E0**	2.59E0	2.54E0	1.35E0	7.66E-1
	MaF06	**1.64E0**	1.17E0	1.05E0	1.6	1.58E0	1.39E0	1.54E0	1.33E0	1.64E0	2.46E	1.55E0
	MaF07	**1.97E0**	1.16E0	1.2E0	1.49E0	1.83E0	2.82E-1	1.96E0	1.94E0	1.26E0	9.95E-1	1E0
10	MaF08	**8.59E-1**	7.61E-1	8.38E-1	6.75E-1	4.26E-1	2.69E-1	5E-1	6.96E-1	8.47E-1	8.54E-1	8.45E-1
	MaF09	1.16E0	7.15E-1	**1.17E0**	3.76E-1	5.89E-1	8.27E-2	6.91E-1	6.48E-1	1.17E0	1.36E-2	4.8E-2
	MaF10	2.47E0	2.59E0	2.53E0	**2.59E0**	2.57E0	2.48E0	2.54E0	2.49E0	2.59E0	1.47E0	2.31E0
	MaF11	2.59E0	2.59E0	2.59E0	2.59	2.59E0	2.55E0	2.59E0	2.59E0	**2.59E0**	2.52E0	2.59E0
	MaF12	2.52E0	2.42E0	2.54E0	1.92E0	2.53E0	2.48E0	**2.54E0**	2.51E0	2.53E0	2.23E0	2.5E0
	MaF13	1.77E0	1.43E0	**1.82E0**	1.69E0	1.02E0	1.39E0	4.77E-1	9.18E-1	1.69E0	1.66E0	1.73E0
	MaF14	2.47E0	1.45E0	**2.58E0**	1.62E0	2.1E0	2.37E0	7.15E-1	5.17E-1	2.32E0	0E0	0E0
	MaF15	1.99E-1	7.55E-2	**2E-1**	1.23E-1	6.38E-2	6.94E-2	2.98E-2	1.12E-1	1.3E-7	0E0	0E0
	MaF01	**4.3E-2**	2.52E-2	3.13E-2	2.74E-2	6.25E-3	6.78E-3	8.11E-3	8.16E-3	3.81E-2	1.83E-2	3.34E-2
	MaF02	3.9E0	3.63E0	**3.95E0**	2.66E0	2.42E0	3.71E0	3.68E0	3.64E0	3.75E0	3.35E0	3.78E0
	MaF03	3.76E0	2.28E0	4.17E0	2.38E0	3.53E0	4.15E0	3.88E0	4.07E0	**4.18E0**	0E0	0E0
	MaF04	**7.39E-2**	2.85E-2	1.57E-2	5.19E-3	9.71E-3	0E0	2.71E-2	3.53E-2	2.57E-2	3.55E-2	6.06E-2
	MaF05	4.06E0	2.83E0	4.15E0	3.36E0	4.09E0	4E0	4.18E0	**4.18E0**	3.9E0	1.64E0	2.67E0
	MaF06	**2.57E0**	7.26E-1	2.07E0	3.83E-1	2.52E0	2.29E0	2.54E0	2.14E0	2.57E0	0E0	2.34E0
	MaF07	2.55E0	9.64E-1	1.83E0	1.04E-1	1.92E0	1.57E-1	**2.83E0**	2.76E0	3.37E-1	3.24E-1	5.96E-1
15	MaF08	**5.72E-1**	4.19E	5.56E-1	4.04E-1	1.09E-1	2.5E-1	2.61E-1	4.07E-1	5.7E-1	5.64E-1	5.33E-1
	MaF09	**9.35E-1**	7.33E-1	9.16E-1	2.86E-1	9.89E-2	1.83E-1	2.81E-1	4.79E-1	8.42E-1	6.85E-1	1.62E-1
	MaF10	**4.18E0**	4.09E0	4.06E0	4.16E0	4.09E0	3.96E0	4.1E0	4.11E0	4.18E0	2.22E0	4.17E0
	MaF11	**4.18E0**	4.03E0	4.16E0	4.17E0	4.15E0	4.09E0	3.87E0	4.17E0	4.18E0	3.99	4.18E0
	MaF12	**4.11E0**	3.69E0	4.04E0	2.69E0	4.06E0	3.88E0	4.09E0	4.07E0	3.81E0	3.03E0	3.98E0
	MaF13	2.6E0	1.37E0	**2.64E0**	2.49E0	9.12E-1	2.22E0	2.74E-1	5.74E-1	2.37E0	1.67E0	2.38E0
	MaF14	3.88E0	8.92E-1	**4.08E0**	1.87E0	3.8E0	3.62E0	2.99E0	2.06E0	3.39E0	0E0	5.6E-1
	MaF15	**2.15E-2**	3.1E-3	1.67E-2	1.72E-2	2.2E-3	6.85E-3	2.39E-3	7.26E-4	0E0	0E0	0E0

effectiveness of HH-CO and the impact of the proposed migration step on hyper-heuristics for

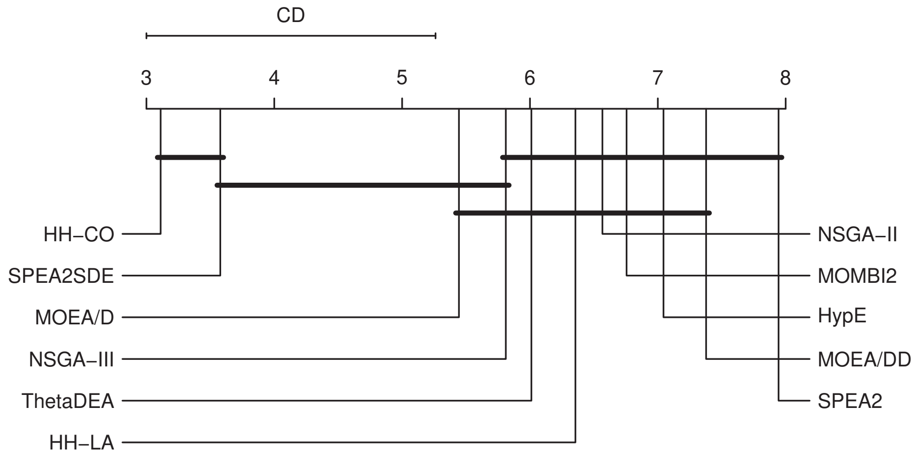


Figure 5.4: Critical Difference plot for hypervolume comparing HH-CO, HH-LA and nine MOEAs

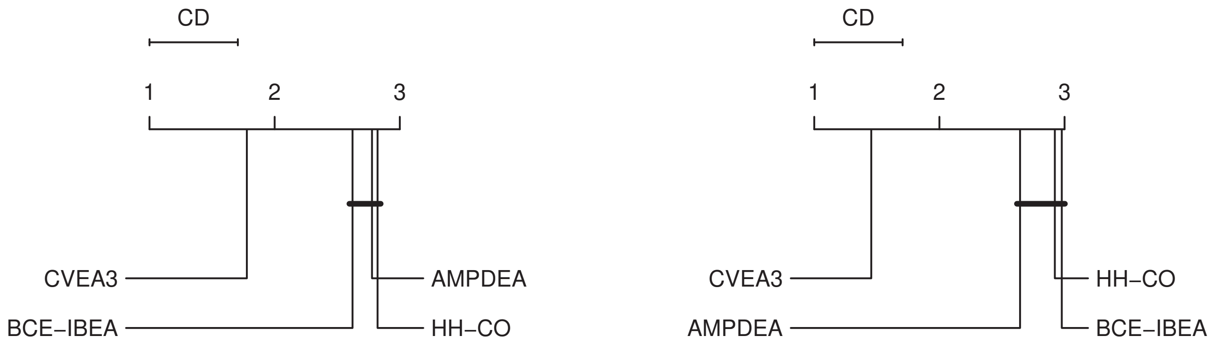
many-objective optimization. Another observation is that the best MOEA per problem instance varies given the diverse characteristics of the benchmark set. However, HH-CO achieved results as good as the best MOEA in most problem instances (95% for IGD and 80% for hypervolume). Those results demonstrate the HH-CO capability of achieving good results in a variety of problem instances. Besides, HH-CO achieved an average ranking better than the best performing MOEA (SPEA2SDE), with statistical significance compared to the other 8 MOEAs.

It was possible to notice that some problem characteristics affected the results. For example, HH-CO performed better on instances with inverted fronts (MaF01, 04, and 15) or with a complicated fitness landscape (MaF04, 10, 12, and 15). Furthermore, the results are more favorable when increasing the number of objectives. On the other hand, the results are less promising in the MaF05 instance. In this problem, HH-CO had worse hypervolume than the best MOEAs for all numbers of objectives. MaF05 is a Convex, biased, badly-scaled DTLZ4. In this problem, the distribution of weight vectors fits the Pareto front shape. This characteristic favors decomposition-based algorithms. For this reason, MOMBI2, ThetaDEA, and NSGA-III usually presented the best results. Besides, they normalize the objective space to deal with badly-scaled fronts.

5.4.3 Comparing the cooperative hyper-heuristic to winners of CEC'18

Next, we compared the proposed HH-CO to the three best-placed algorithms from the Competition on Many-Objective Optimization at the 2018 IEEE Congress on Evolutionary Computation (CEC). The algorithms are CVEA3, AMPDEA, and BCE-IBEA. First, the analysis of the MOEAs demonstrates that CVEA3 achieved better results than AMPDEA and BCE-IBEA, in both IGD and HV, with statistical significance. AMPDEA and BCE-IBEA were statistically equivalent in both IGD and HV (Figure 5.5). When included HH-CO, it ranked closely to AMPDEA (IGD) and BCE-IBEA (hypervolume). Moreover, the statistical analysis demonstrates that HH-CO was equivalent to the second place algorithm on both indicators.

We also compared the results per problem instance. When evaluating the IGD (Table 5.9), HH-CO achieved the best average value in two instances, MaF13 (10 number of objectives) and MaF07 (15 number of objectives). Besides, it was equivalent to the best in MaF02 (10 and 15 number of objectives), MaF03 (15 number of objectives), MaF09 (10 number of objectives) and MaF13 (5 and 15 number of objectives). The main difficulty in MaF02 and MaF13 is convergence. Another observation is that the best MOEA per problem varies among CVEA3 (19



(a) IGD

(b) Hypervolume

Figure 5.5: Critical Difference plot comparing HH-CO and winners from CEC'18

Table 5.9: Average values for IGD comparing HH-CO and winners from CEC'18

Obj.	problem	HH-CO	CVEA3	AMPDEA	BCE-IBEA
5	MaF01	1.27E-3	1.19E-3	1.16E-3	**1.14E-3**
	MaF02	6.92E-3	6.71E-3	**6.29E-3**	6.53E-3
	MaF03	1.3E-2	**6.3E-4**	9.03E-4	2.58E-3
	MaF04	2.05E-2	**1.85E-2**	1.98E-2	2.39E-2
	MaF05	2.3E-2	1.98E-2	3.3E-2	**1.89E-2**
	MaF06	3.35E-5	2.39E-5	4.13E-5	**2.22E-5**
	MaF07	2.72E-3	**2.22E-3**	2.75E-3	2.3E-3
	MaF08	1.12E-3	1.04E-3	1.19E-3	**9.71E-4**
	MaF09	1.35E	**1.05E-3**	1.57E-3	4.54E-3
	MaF10	7E-3	4.59E-3	2.3E-2	**3.94E-3**
	MaF11	7.62E-3	**6.53E-3**	1E-2	7.35E-3
	MaF12	1.12E-2	1.03E-2	2.01E-2	**1E-2**
	MaF13	9.16E-4	**9.02E-4**	3.91E-3	1.25E-3
	MaF14	6.39E-3	3.79E-3	**2.54E-3**	8.51E-3
	MaF15	2.97E-3	**2.26E-3**	4.02E-3	1.28E-2
10	MaF01	2.85E-3	2.98E-3	**2.76E-3**	2.84E-3
	MaF02	2.06E-3	2.06E-3	2.19E-3	**2.06E-3**
	MaF03	1.09E-1	2.16E-3	**1.17E-3**	1.44E8
	MaF04	7.65E-1	6.49E-1	**6.11E-1**	1.35E0
	MaF05	1.8E0	7.35E-1	9.36E-1	**6.92E-1**
	MaF06	1.96E-4	**2.29E-5**	6.87E-5	5.6E-3
	MaF07	7.23E-3	**6.06E-3**	6.85E-3	6.18E-3
	MaF08	1.6E-3	1.55E-3	2.22E-3	**1.33E-3**
	MaF09	3.03E-3	**2.47E-3**	8.52E-3	3.18E-2
	MaF10	1.82E-2	1.66E-2	2.99E-2	**1.26E-2**
	MaF11	2.31E-2	**1.77E-2**	2.38E-2	1.98E-2
	MaF12	5.44E-2	**5.03E-2**	6.94E	5.05E-2
	MaF13	**1.44E-3**	1.44E-3	4.45E-3	1.5E-3
	MaF14	8.34E-3	5.92E-3	**4.98E-3**	2.89E-2
	MaF15	8.97E-3	**6.29E-3**	6.41E-3	6.51E-2
15	MaF01	3.9E-3	4.2E-3	**3.71E-3**	4.01E-3
	MaF02	2.41E-3	2.49E-3	**2.35E-3**	3.69E-3
	MaF03	2.63E-2	2.48E-3	**1.37E-3**	1.12E9
	MaF04	2.31E1	2.27E1	**1.91E1**	4.49E1
	MaF05	3.88E1	2.29E1	2.83E1	**2.12E1**
	MaF06	3.21E	**2.36E-5**	9.76E-5	6.4E-2
	MaF07	**1.28E-2**	1.3E-2	1.36E-2	1.87E-2
	MaF08	1.97E-3	1.99E-3	2.12E-3	**1.64E-3**
	MaF09	3.39E-3	**1.92E-3**	1.45E-2	6.21E-3
	MaF10	2.62E-2	3.02E-2	3.77E-2	**2.19E-2**
	MaF11	1.01E-2	**2.7E-5**	3.72E-2	5.44E-2
	MaF12	1.09E-1	**9.47E-2**	1.45E-1	9.61E-2
	MaF13	1.73E-3	**1.58E-3**	4.03E-3	1.65E-3
	MaF14	1.09E-2	8.64E-3	**6.27E-3**	1.58E-1
	MaF15	1.25E-2	**8.03E-3**	1.11E-2	7.92E-2

times), AMPDEA (11 times), and BCE-IBEA (13 times). CVEA3 presented an average IGD worse (with statistical significance) than the best MOEA in 18 out of 45 instances (40%). These observations demonstrate that even the best MOEA for the benchmark does not present good results in all problem instances.

When evaluating the hypervolume, the conclusions are similar. HH-CO achieved the best average hypervolume in two instances: MaF10 and MaF12, with a 15 number of objectives. Besides, it was statistically equivalent to the best in other 7 instances: MaF02 (15), MaF03 (10 and 15 objectives), MaF06 (10 objectives), MaF09 (10 and 15 objectives) and MaF13 (5 objectives). Compared to other MOEAs, CVEA3 achieved the best hypervolume average on 2/3 of the problem instances.

Table 5.10: Average values for hypervolume comparing HH-CO and winners from CEC'18

Obj.	problem	HH-CO	CVEA3	AMPDEA	BCE-IBEA
5	MaF01	9.88E-4	1.13E-3	**1.37E-3**	1.26E-3
	MaF02	3.5E-1	3.93E-1	3.93E-1	**3.95E-1**
	MaF03	8.42E-1	**9.99E-1**	9.98E-1	9.77E-1
	MaF04	4.2E-2	**7.08E-2**	6.09E-2	3.59E-2
	MaF05	6.12E-1	**6.96E-1**	5.55E-1	6.92E-1
	MaF06	3.48E-2	3.47E-2	3.48E-2	**3.5E-2**
	MaF07	1.73E-1	2.3E-1	2.3E-1	**2.33E-1**
	MaF08	8.18E-2	**8.46E-2**	8.37E-2	8.33E-2
	MaF09	2.34E-1	**2.42E-1**	2.25E-1	1.64E-1
	MaF10	9.25E-1	**9.98E-1**	5.75E-1	9.95E-1
	MaF11	9.94E-1	**9.97E-1**	9.77E-1	9.94E-1
	MaF12	5.81E-1	**6.8E-1**	4.77E-1	6.33E-1
	MaF13	2.37E-1	**2.4E-1**	8.04E-2	2.11E-1
	MaF14	2.79E-1	6.98E-1	**7.17E-1**	2.15E-1
	MaF15	1.09E-2	**4.31E-2**	1.65E-2	0E0
10	MaF01	2.18E-1	**2.51E-1**	2.36E-1	1.52E-1
	MaF02	2.43E0	**2.46E0**	2.45E0	2.45E0
	MaF03	2.46E0	**2.59E0**	2.59E0	1.89E0
	MaF04	3.72E	4.75E-1	**4.85E-1**	2.89E-1
	MaF05	2.49E0	**2.59E0**	2.59E0	2.59E0
	MaF06	1.64E0	1.64E0	**1.64E0**	1.36E0
	MaF07	1.97E0	**2.13E0**	1.99E0	2.08E0
	MaF08	8.6E-1	8.64E-1	8.42E-1	**8.64E-1**
	MaF09	1.16E0	**1.16E0**	8.08E-1	2.45E-1
	MaF10	2.47E0	**2.59E0**	2.48E0	2.59E0
	MaF11	2.59E0	2.59E0	**2.59E0**	2.59E0
	MaF12	2.54E0	**2.57E0**	2.43E0	2.53E0
	MaF13	1.77E0	**1.81E0**	8.36E-1	1.74E0
	MaF14	2.47E0	**2.59E0**	2.34E0	1.09E0
	MaF15	1.99E-1	3.5E-1	**3.51E-1**	1.75E-2
15	MaF01	4.65E-2	**5.33E-2**	4.8E-2	1.76E-2
	MaF02	3.94E0	**3.95E0**	3.94E0	3.95E0
	MaF03	3.76E0	**4.18E0**	4.18E0	2.56E0
	MaF04	9.37E-2	1.29E-1	**1.38E-1**	4.6E-2
	MaF05	4.1E0	4.18E0	**4.18E0**	4.18E0
	MaF06	2.57E0	**2.58E0**	2.58E0	1.52E0
	MaF07	2.71E0	**3.15E0**	2.96E0	3.05E0
	MaF08	5.88E-1	**5.96E-1**	5.92E-1	5.95E-1
	MaF09	9.43E-1	**9.81E-1**	4.35E-1	8.23E-1
	MaF10	**4.18E0**	4.18E0	3.99E0	4.17E0
	MaF11	4.18E0	4.18E0	**4.18E0**	4.17E0
	MaF12	**4.13E0**	4.01E0	3.95E0	4.05E0
	MaF13	2.6E0	**2.68E0**	1.12E0	2.49E0
	MaF14	3.9E0	**4.18E0**	3.86E0	2.43E-1
	MaF15	2.34E-2	**9.33E-2**	3.86E-2	1.55E-3

The analysis of the results demonstrates that HH-CO was competitive to the second and third best MOEAs from the CEC'18 competition. On the other hand, the results were not competitive to the best MOEA on MaF benchmark (CVEA3). Finally, HH-CO results were more favorable in problems where the difficulty was convergence. Besides, the results are improved when increasing the number of objectives. It is worth noticing that those algorithms are not in the pool of MOEAs used by HH-CO. Therefore, HH-CO results would possibly be better if it had the chance of choosing from CVEA3, BCE-IBEA, and AMPDEA as well. However, the analysis presented reveals how the HH-CO (with its current pool of MOEAs) interacts with the best algorithms known for this problem set.

5.4.4 Comparing the two proposed cooperative hyper-heuristics

Additionally, we compared the two proposed cooperative hyper-heuristics, HH-CO, and its former, the HHcMOEA. For this comparison, we also used the MaF benchmark and observed both IGD and hypervolume. On the IGD analysis, demonstrated in Table 5.11, it is possible to see that the HH-CO presented better average IGD on 32 problems instances (71.11%), against 13 for HHcMOEA (28.89%). In the overall IGD analysis, performed by the Friedman rank-sum test, we found a p-value = 0.004621, which is smaller than the 0.05 threshold. Therefore, it means that HH-CO and HHcMOEA medians are not equal. Finally, the average ranking for HH-CO was 1.29, while HHcMOEA was 1.71, being 1.0 and 2.0 the best and worst possible average ranking values. In detail, the HH-CO advantage was higher for a smaller number of objectives. That is, it was better on 13 instances for five objectives (86.67%), 10 instances for ten objectives (66.67%), and 9 instances for fifteen objectives (60%). The HH-CO disadvantage was mainly on MaF15 (inverted Pareto front and complicated fitness landscape) and MaF03 (convex DTLZ3 with a large number of local fronts).

On the hypervolume analysis (Table 5.12), the observations are similar to IGD. However, the degree of the advantage of HH-CO over HHcMOEA reduces. The HH-CO achieved better hypervolume on 28 problems (62.22%), with an overall average ranking of 1.36 against 1.64 from HHcMOEA. On the statistical test we found a p-value = 0.04743. It is possible to say that HH-CO and HHcMOEA medians are not equal (considering a 0.05 threshold). In detail, HH-CO achieved better hypervolume values for 10 instances for five objectives, 10 intances for ten objectives, and 8 for fifteen objectives.

The analysis of the results demonstrates that the HH-CO improves the quality of HHcMOEA, mainly when observed in the IGD. However, as the number of objectives increases, the problems become more complicated. Then, the difference between HH-CO and HHcMOEA diminishes. The differences between HH-CO and HHcMOEA frameworks are presented in Section 4.5.

5.4.5 Discussion

The evaluation of HH-CO, in this section, followed the same benchmark and methodology used in the competition on many-objective from CEC 2018. First, we compared HH-CO to the MOEAs from the pool and a state-of-the-art hyper-heuristic HH-LA. The results were favorable to HH-CO, which achieved the best or equivalent to the best result in most problem instances. Regarding IGD, it was the best or equivalent to the best in 96% of the problem instances, while for hypervolume, it was the best or equivalent in 80% of the problem instances. HH-CO presented good results mainly for challenging problems, including inverted fronts and complicated fitness landscapes. Besides, the results were better when increasing the number of objectives. However, it is worth noticing that HH-LA has initially been proposed and evaluated, with a fine-tuning of its parameters, for three objectives. Therefore, an adequate parameter tuning for HH-LA could improve its results for many-objective optimization. On the other hand, HH-CO has no parameters to be set.

Table 5.11: Average (and standar deviation) values for IGD comparing HH-CO and HHcMOEA

Obj.	problem	HH-CO	HHcMOEA
	MaF01	**1.16E-3(1.29E-5)**	1.44E-3(5.23E-5)
	MaF02	**6.45E-3(7.95E-5)**	7.49E-3(2.36E-4)
	MaF03	1.39E2(2.04E2)	**1.47E-3(2.25E-4)**
	MaF04	**1.79E-2(2.22E-4)**	2.82E-2(1.72E-2)
	MaF05	**1.86E-2(4.22E-4)**	3E-2(3.94E-3)
	MaF06	**3.86E-5(1.27E-6)**	7.77E-5(2.59E-5)
	MaF07	**2.21E-3(4.37E-5)**	2.78E-3(1.63E-4)
5	MaF08	**1E-3(1.08E-5)**	1.46E-3(7.9E-5)
	MaF09	**1.02E-3(5.07E-5)**	1.53E-3(1.27E-4)
	MaF10	**8.36E-3(7.79E-4)**	1.03E-2(3.59E-3)
	MaF11	**8.86E-3(2.16E-3)**	9.38E-3(1.12E-3)
	MaF12	**9.93E-3(5.66E-5)**	1.24E-2(8.37E-4)
	MaF13	**8.54E-4(8.16E-5)**	1.21E-3(1.34E-4)
	MaF14	**6.07E-3(1.38E-3)**	6.32E-3(8.24E-4)
	MaF15	8.78E-3(1.2E-3)	**4.88E-3(2.36E-3)**
	MaF01	**2.76E-3(1.85E-5)**	3.42E-3(1.02E-4)
	MaF02	**2.03E-3(2.95E-5)**	2.15E-3(7.92E-5)
	MaF03	1.73E3(1.91E3)	**1.92E-3(2.59E-4)**
	MaF04	**7.08E-1(2.26E-2)**	1.37E0(1.55E0)
	MaF05	**1.01E0(7.64E-2)**	1.39E0(2.26E-1)
	MaF06	1.76E-2(1.41E-2)	**2.83E-3(3.03E-3)**
	MaF07	**6.66E-3(1.51E-4)**	8.17E-3(1.33E-3)
10	MaF08	**1.38E-3(1.05E-5)**	1.92E-3(5.38E-5)
	MaF09	1.79E-2(5.47E-3)	**4.85E-3(6.34E-4)**
	MaF10	**1.9E-2(1.8E-3)**	2.36E-2(8.15E-3)
	MaF11	2.7E-2(3.05E-3)	**2.5E-2(1.57E-2)**
	MaF12	**5.22E-2(9.21E-4)**	5.68E-2(4.08E-3)
	MaF13	**1.07E-3(9.01E-5)**	2.03E-3(2.13E-4)
	MaF14	**1.06E-2(2.48E-3)**	2.65E-2(6.31E-2)
	MaF15	5.51E-2(4.57E-2)	**2.37E-2(2.52E-2)**
	MaF01	**3.58E-3(5.06E-5)**	4.83E-3(2.8E-4)
	MaF02	**2.34E-3(3.08E-5)**	3.12E-3(2.09E-4)
	MaF03	1.06E-1(1.97E-1)	**2.3E-3(1.31E-4)**
	MaF04	**2.27E1(1.6E0)**	1.66E2(2.89E2)
	MaF05	**2.91E1(2.04E0)**	4.87E1(7.3E0)
	MaF06	8.63E-3(7.84E-3)	**4.27E-4(2.88E-4)**
	MaF07	**1.18E-2(1.47E-4)**	1.43E-2(4.4E-4)
15	MaF08	2.11E0(5.49E0)	**2.24E-3(4.18E-5)**
	MaF09	**1.96E-3(9.97E-5)**	4.94E-3(1.47E-3)
	MaF10	**2.48E-2(9.67E-4)**	3.17E-2(5.74E-3)
	MaF11	6.16E-2(1.81E-2)	**4.17E-2(2.03E-2)**
	MaF12	**1.03E-1(1.44E-3)**	1.24E-1(4.09E-3)
	MaF13	**1.35E-3(1.13E-4)**	2.19E-3(2.23E-4)
	MaF14	1.35E-2(1.97E-3)	**1.11E-2(1.97E-3)**
	MaF15	6.47E-2(5.85E-2)	**2.95E-2(2.19E-2)**

Table 5.12: Average (and standar deviation) values for hypervolume comparing HH-CO and HHcMOEA

Obj.	problem	HH-CO	HHcMOEA
	MaF01	**1.24E-2(2.07E-4)**	8.64E-3(5.56E-4)
	MaF02	**2.03E-1(1.28E-3)**	1.73E-1(5.22E-3)
	MaF03	0E0(0E0)	**9.96E-1(1.75E-3)**
	MaF04	**1.14E-1(3.45E-3)**	7.07E-2(2.14E-2)
	MaF05	**7.78E-1(3.65E-3)**	6.9E-1(3.45E-2)
	MaF06	**1.3E-1(3.59E-4)**	1.29E-1(1.02E-3)
	MaF07	**2.62E-1(1.7E-3)**	2.39E-1(1.3E-2)
5	MaF08	**1.26E-1(4.42E-4)**	1.2E-1(6.9E-4)
	MaF09	**3.27E-1(2.38E-3)**	3.16E-1(4.26E-3)
	MaF10	7.31E-1(3.26E-2)	**9.65E-1(4.11E-2)**
	MaF11	9.92E-1(1.01E-3)	**9.94E-1(2.08E-3)**
	MaF12	**7.57E-1(4.17E-3)**	6.84E-1(2.2E-2)
	MaF13	**2.93E-1(4.29E-3)**	2.76E-1(6.7E-3)
	MaF14	4.08E-1(1.51E-1)	**4.53E-1(1.09E-1)**
	MaF15	6.21E-4(1.6E-3)	**2.36E-2(1.76E-2)**
	MaF01	**5E-7(6.88E-7)**	1E-7(3.08E-7)
	MaF02	**2.04E-1(2.26E-3)**	1.69E-1(6.42E-3)
	MaF03	0E0(0E0)	**1E0(3.01E-4)**
	MaF04	**1.52E-4(1.97E-5)**	2.07E-5(2.08E-5)
	MaF05	**7.02E-1(3.69E-2)**	2.52E-1(1.47E-1)
	MaF06	1.4E-2(3.13E-2)	**9.22E-2(9.29E-3)**
	MaF07	**7.4E-2(1.87E-2)**	4.63E-2(3.22E-2)
10	MaF08	**1.05E-2(1.11E-4)**	9.77E-3(1.55E-4)
	MaF09	5.96E-4(2.66E-4)	**9.91E-3(1.22E-3)**
	MaF10	6.53E-1(7.41E-2)	**9.96E-1(6.92E-3)**
	MaF11	**9.99E-1(3.4E-4)**	9.97E-1(2.07E-3)
	MaF12	**7.63E-1(3.62E-2)**	6.82E-1(8E-2)
	MaF13	**1.37E-1(1.83E-3)**	1.16E-1(4.14E-3)
	MaF14	**2.19E-1(1.54E-1)**	2.15E-1(2.48E-1)
	MaF15	0E0(0E0)	**5E-8(2.24E-7)**
	MaF01	0E0(0E0)	**0E0(0E0)**
	MaF02	1.53E-1(8.3E-3)	**1.71E-1(6.33E-3)**
	MaF03	5.9E-1(4.79E-1)	**1E0(6.39E-5)**
	MaF04	**1.5E-7(3.66E-7)**	5E-8(2.24E-7)
	MaF05	**7.42E-1(6.59E-2)**	3.97E-1(8.36E-2)
	MaF06	1.34E-2(2.73E-2)	**9.2E-2(7.13E-3)**
	MaF07	**3.47E-2(1.54E-2)**	1.57E-3(3.18E-3)
15	MaF08	4.37E-4(2.59E-4)	**5.31E-4(1.9E-5)**
	MaF09	**1.18E-3(4.69E-5)**	6.15E-4(1.53E-4)
	MaF10	**1E0(5.61E-5)**	1E0(4.69E-4)
	MaF11	**1E0(1.45E-4)**	9.99E-1(8.29E-4)
	MaF12	**8.08E-1(3.84E-2)**	6.71E-1(5.16E-2)
	MaF13	**8.77E-2(1.55E-3)**	7.73E-2(2.97E-3)
	MaF14	1.2E-1(9.37E-2)	**3.26E-1(1.9E-1)**
	MaF15	0E0(0E0)	**0E0(0E0)**

Finally, we also compared HH-CO to the winner algorithms on the CEC'18 competition. The results were statistically equivalent to the second and third best algorithms. In this case, the results of HH-CO also improved when increasing the number of objectives. Moreover, the best MOEA from the competition winners varies for different problem instances. Thus, it is likely that HH-CO would have achieved a better overall result, including those MOEAs into its pool[5]. Moreover, we demonstrated that the HH-CO improves the results from its former version, the HHcMOEA.

[5]The current MATLAB implementation is not compatible with the jMetal framework used in this research.

5.5 EMPIRICAL ANALYSIS OF THE WIND TURBINE DESIGN PROBLEM

This section presents an empirical analysis of eight state-of-the-art MOEAs and the HH-CO on a recently proposed many-objective problem. In this study, the HH-CO reward, which uses the R2 metric improvement, was also adapted to consider the constraint violations (using the decomposition-based strategy). Further, HH-CO uses: NSGA-II, SPEA2, ThetaDEA, NSGA-III, MOMBI2, SPEA2SDE, HypE, and MOEA/D. It is important to note that the solution comparison method of all algorithms was modified to handle constraints. However, the MOEA/DD was not used since, in the preliminary analysis, it presented difficulties in finding feasible solutions. The MOEA/DD is based on both Pareto dominance and decomposition. However, in the Pareto dominance comparison, the MOEA/DD may prefer a dominated solution to preserve diversity. On the constraint handling approach used, the unfeasible solutions are said dominated by the feasible ones. However, the characteristics of MOEA/DD causes it to sometimes prefer the unfeasible (dominated) ones.

The problem instance is the Wind Turbine Design problem (The Japanese Society of Evolutionary Computation, 2019), with five objectives, 32 variables and 22 constraints, detailed in Section 2.1.9. A module, publicly available, evaluates the objective functions and constraints. The maximum number of fitness evaluations (FE) is $10,000$, and the population size is 210 for all algorithms (Deb and Jain, 2014). That yields 47 iterations ($9,870$ FE), including the random initialization of the populations. Finally, the hypervolume (HV) quality indicator evaluates the results (see Section 2.1.6), using the reference point given by the competition methodology. Besides, the HV values presented are normalized and take only the feasible solutions that dominate the reference point.

For the empirical analysis, we evaluated the hypervolume during the search of the eight MOEAs and the HH-CO. The hypervolume is computed over an unbounded repository of non-dominated solutions. In other words, every time one generates a new solution, its repository is updated and the HV computed. In this analysis, when the hypervolume equals zero, no feasible solution dominates the reference point (probably, all solutions found so far are infeasible). Next, we present two analyses: first, the median HV trial (considering the external repository at the end of the search) of each MOEA and HH-CO. Then, we present an analysis of the overall behavior for the 21 runs. This analysis is different from the one used in previous experiments due to the number of analyzed problems. This experiment details the behavior of the algorithms on a single problem differently than the earlier experiments, which demonstrated the overall behavior of the algorithms on a set of problems. Therefore, the analysis focused on just one problem, allowed us to look inside the convergence of HH-CO and the MOEAs during the search, and the choices made by HH-CO.

5.5.1 Analysis for the median run

At Figure 5.6, the dashed line shows the HV for HH-CO, and the other lines the HV of every MOEA. First, the best performing MOEA, considering the median HV, was NSGA-III, the second-best MOEA was ThetaDEA. Note that it performed worse than most MOEAs during most of the search (from FE $2,000$ to FE $7,000$). Then, SPEA2 and SPEA2SDE achieved similar HV at the end. However, SPEA2 was the slowest MOEA to start converging (about $2,500$ FE). On the other hand, SPEA2SDE was one of the best performing MOEAs in the beginning (from FE $2,000$ to FE $6,000$). Furthermore, NSGA-II and MOMBI2 were no better or worse than other MOEAs, considering the entire search. Finally, MOEA/D was the best MOEA up to FE $4,000$ (the MOEA/D is the only MOEA in the pool to use Differential Evolution instead of the SBX crossover). Still, it stagnated and finished with the second worse result, followed only by HypE.

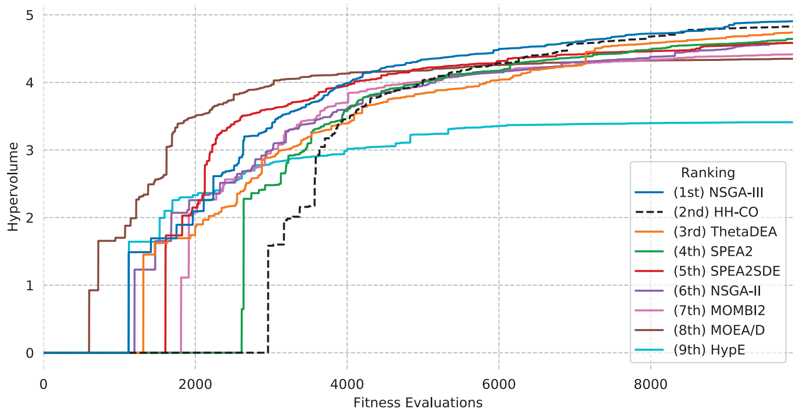


Figure 5.6: The median hypervolume of an external repository of non-dominated solutions for the HH-CO (dashed line) and every MOEA (other lines) during the search.

In this analysis, the HH-CO achieved the second-best hypervolume, worse than only NSGA-III. It is noticeable that HH-CO took longer than any MOEA to start converging since it generates a set of solutions with random values for each MOEA. Therefore, it took $8 \times 210 = 1,680$ FE only to initialize. After that, it converged and had results better than most MOEAs from FE $6,000$ onwards. The HV values of every MOEA during the search can help us understand the choices of the HH-CO at different stages of the search. For example, it is possible to suggest which MOEAs it should select more often, according to how the MOEAs behaved individually.

Figure 5.7, presents the choices made by the HH-CO during the search. It makes its first choice after $1,680$ FE (i.e., after the initialization). We can observe that ThetaDEA was the most selected MOEA (8 times). ThetaDEA and SPEA2 were the most selected from the middle to the end of the search. That is, when they perform better, as we have seen in hypervolume analysis. Likewise, SPEA2SDE and NSGA-II were the most selected from the beginning to the middle of the search. The HH-CO selected MOMBI2, NSGA-III, and HypE uniformly distributed over the search. Finally, MOEA/D was the less selected MOEA. The diversity mechanism of SPEA2SDE (that considers the distribution and the convergence of solutions) could explain the preference for it at the beginning. On the other hand, the application of ThetaDEA, with $\theta = 5$, (Ishibuchi et al., 2017) could provide more convergence at the end (a preference for exploitation at the final iterations).

It is noticeable that the MOEA with the best hypervolume when applied alone (NSGA-III), was not among the most selected MOEAs by HH-CO. A possible explanation is the greedy selection method used. It seeks to find MOEAs with significant improvements rather than one with small but consistent improvements. Moreover, the HH-CO changes the applied MOEA every iteration. Additionally, as knowledge is kept internally safe, the wrong choices are not critical for the search. Finally, we can notice that HH-CO selected all MOEAs at least once. In other words, every MOEA improved more than all others at a given point in the search. That points out the importance of diversity in the set of MOEAs used by HH-CO.

5.5.2 Overall analysis

Figure 5.8 presents the boxplot analysis for hypervolume achieved by every MOEA and HH-CO. This analysis supports the critical difference plot of 21 runs, presented in Figure 5.9. It is possible to observe that the HV of the MOEAs highly varies from the best to the worst. We can also see

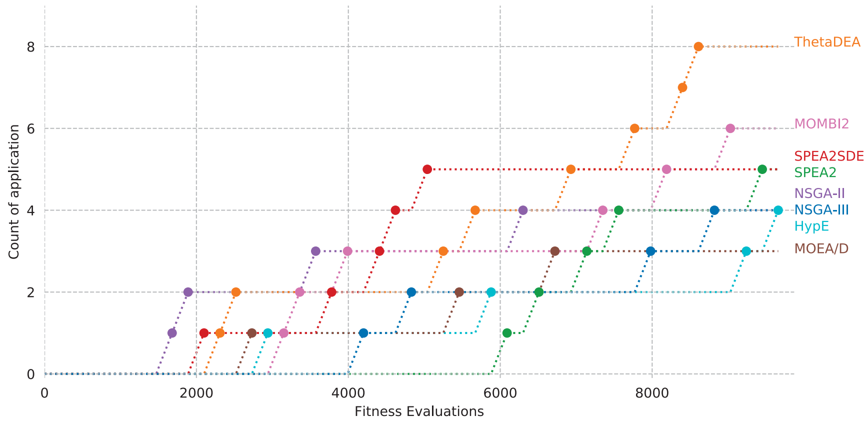


Figure 5.7: The count of how many times HH-CO applied each MOEA. It was selected the median HV trial for this analysis.

that the two best MOEAs are from those specifically designed for many-objective optimization (NSGA-III and ThetaDEA). In general, the best-performing algorithms for this problem were NSGA-III, followed by the hyper-heuristic (HH-CO), and ThetaDEA. From those three, the statistical test did not find a significant difference. From the boxplot, we observe that the HH-CO was the one with the smallest variability in the results among different runs. Besides, SPEA2 and NSGA-II, Pareto-based MOEAs, achieved competitive results without a significant difference to the state-of-the-art MOEAs like ThetaDEA and SPEA2SDE. Finally, the worst-performing MOEAs for this problem were MOEA/D, MOMBI2, and HypE. Although the boxplot shows differences, the statistical test did not significantly distinguish between HypE and the other four worse-performing MOEAs.

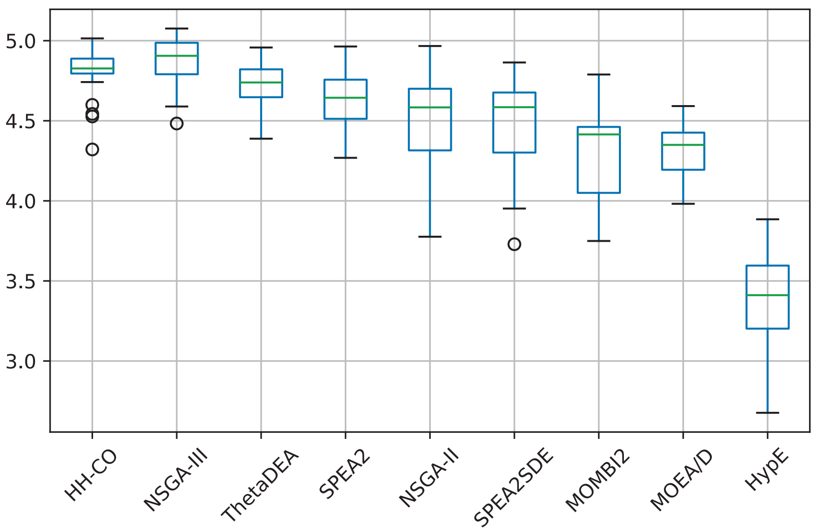


Figure 5.8: Boxplot of hypervolume for HH-CO and each MOEA after 21 runs.

Figure 5.10 presents the accumulated number of applications for every MOEA. In general, HH-CO applied each MOEA about five times. SPEA2 and SPEA2SDE stand out, executed around 7.5 times each. On the other hand, NSGA-II was the less applied MOEA. Since

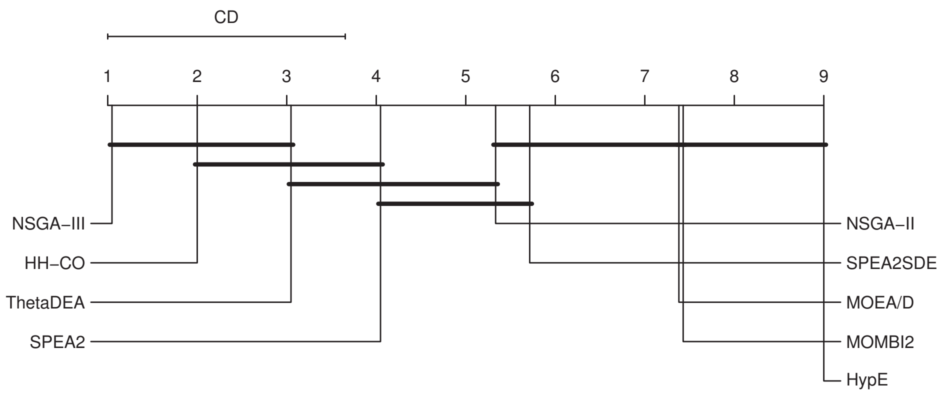


Figure 5.9: Critical difference plot of Hypervolume. Each algorithm is connected to its average ranking from 21 trials, and a bold horizontal line connects algorithms without significant statistical difference

each MOEA performs differently during the search, HH-CO selects them at different search phases. However, most of them end up having a similar accumulated number of applications.

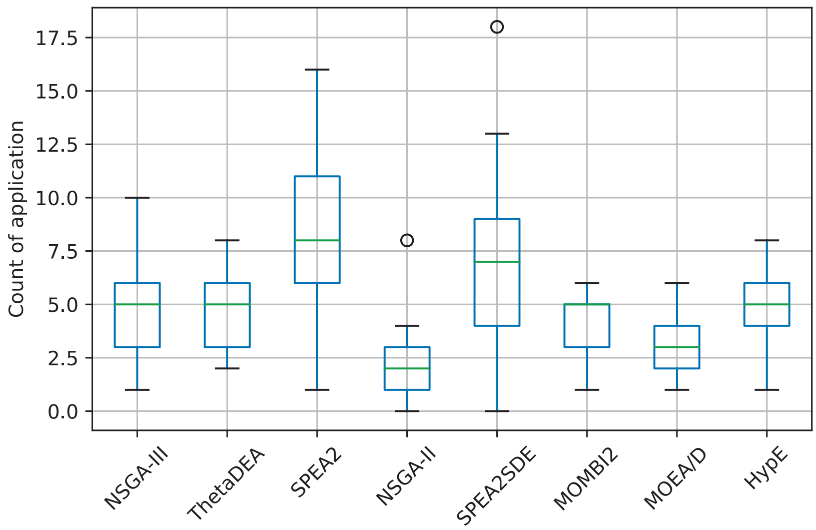


Figure 5.10: Boxplot of the accumulated count of how many times HH-CO applied each MOEA, for 21 runs.

Finally, we present the boxplot with the average execution time, 21 trials, for every MOEA and the HH-CO in Figure 5.11. The experiments ran on an Intel Xeon CPU E5-2640 v3 of 2.60GHz with 32 CPUs and 94GB of RAM. The jobs were launched by a process queue every time the load average drops below 15. The MOEAs spent 16.5 to 18 hours (17.3 hours in average), being ThetaDEA the slowest. The HH-CO took about 19.5 to 20 hours (about 2.5 hours more than a MOEA in average). It means that, for this problem, running HH-CO once is about 14% slower than running an off-the-shelf MOEA. The HH-CO highest cost is the migration step. In this step, it executes the environmental selection method of each MOEA. Therefore, considering this method as the most costly, the HH-CO would take N times longer than a single MOEA (ignoring fitness evaluations), where N the number of MOEAs. In other words, as large the pool of MOEAs, the higher the computational cost of HH-CO.

In summary, the HH-CO is slower than an off-the-shelf MOEA. However, it is usually unknown a priori the best MOEA for a problem. If we want to find the best MOEA for this problem, running every MOEA once will take more than five days. Therefore, using HH-CO is

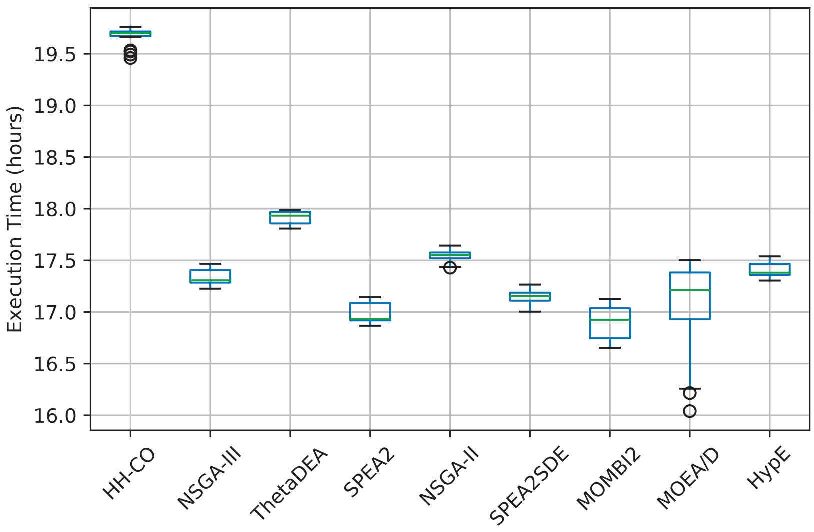


Figure 5.11: Boxplot for the computational time spent by each MOEA and HH-CO

faster than running all MOEAs to find the best one. It is worth noticing that the computational cost of HH-CO can be reduced by decreasing the number of MOEAs in the pool or using communication topologies other than broadcast.

Moreover, as much expensive is the fitness evaluation, the smaller is the difference between HH-CO and the MOEAs. Furthermore, the implementation is the same for the MOEA applied inside the HH or alone. Therefore, the difference in the computation cost of HH-CO and MOEAs is not related to implementation details.

5.5.3 Discussion

In this analysis, we have evaluated a recently proposed real-world many-objective problem. This problem presents interesting properties: 32 continuous variables, five objectives, and 22 constraints. It is simple to reproduce since the fitness evaluation module is publicly available. Moreover, it can serve as a baseline for the comparison of MOEAs. Based on those characteristics, we solved this problem with a set of eight state-of-the-art MOEAs for many-objective optimization — adapted to incorporate constraint handling.

The MOEAs presented a significant difference in quality measured by the hypervolume quality indicator. Moreover, it was possible to identify that some MOEAs had a better HV value at the beginning of the search. Others stagnated after some iterations, while others took a while longer to start converging but keep converging during the rest of the search. Finally, the best-performing MOEAs for this problem were NSGA-III and ThetaDEA. Both are designed explicitly for many-objective optimization. Besides, SPEA2 presented good overall results.

Next, we applied the proposed hyper-heuristic for many-objective optimization, the Cooperative Hyper-heuristic (HH-CO). The results achieved by HH-CO were competitive to the best MOEA. Further, we evaluated the choices made by HH-CO. In general, the preferences agree with the best performing MOEAs in different phases of the search. Finally, we assessed the computational cost. The hyper-heuristic was slower than an average MOEA but with higher quality in terms of hypervolume. Another highlight is that HH-CO took much longer to start convergence since it initializes a different population for each MOEA. One suggestion to overcome this difficulty is to initialize a single population and replicate it for all MOEAs, for problems where the number of fitness evaluations is small.

6 CONCLUSIONS AND FUTURE WORKS

In this research, we proposed the cooperation of multiple MOEAs for solving many-objective problems. The main idea is to execute several MOEAs on the same problem instance. The different characteristics of the MOEAs, working together, would allow good results on a set of problems with different properties. The central aspect of this analysis is the exchange of information during the execution. The use of information from other MOEAs may improve diversity and exploration during the search.

Initially, we presented a heterogeneous distribution of MOEAs, implemented as islands communicating among themselves. The evaluation of this framework for the cooperation of two MOEAs demonstrated the capabilities of the information exchange in improving the search ability. Moreover, the synchronous and asynchronous communication assessment revealed better results for the synchronous version, maximizing the information exchange. However, further experiments demonstrated the inability of this framework to increase the number of MOEAs and other difficulties.

After extensive research and experimental validation, we presented a hyper-heuristic framework based on the cooperation of MOEAs for many-objective optimization (HHcMOEA). In this framework, every MOEA is independent and communicates at every iteration. This communication is needed to keep the information of MOEAs updated during the search process. The hyper-heuristic framework was evaluated with and without the proposed migration mechanism. The results were favorable, with a statistically significant difference, to the version using the migration strategy. Also, it achieved better results, with a significant difference, to the MOEAs from its pool applied standalone.

Finally, an improved version is proposed (HH-CO), incorporating knowledge acquired with experimentation and validation. For instance, the new version exchanges all newly generated solutions instead of the updated population. Moreover, all MOEAs are rewarded (not only the applied one) as they may improve receiving external solutions. Also, we proposed a credit assignment and selection strategy that favors MOEAs that are improving rather than stagnated ones. However, those MOEAs are not excluded from the pool, as they may be useful later. This version is favorable compared to a state-of-the-art hyper-heuristic for multi-objective optimization, with a significant difference. Also, it achieved the best average ranking for both IGD and hypervolume quality indicators. It was compared to the nine MOEAs that compose its pool, with a significant difference to eight of them. We used problems from the CEC'18 many-objective competition. Therefore, we compared the proposed framework to the three best-ranked MOEAs from the competition. The proposed approach achieved results competitive to two of them. Finally, the comparison of HH-CO and HHcMOEA demonstrated that the HH-CO version improves the results from its predecessor.

At last, we evaluated the cooperative hyper-heuristic on a real-world application, the wind-turbine design problem. It achieved results competitive to the best performing MOEAs. In the analysis of the computational cost, it was slower than an average off-the-shelf MOEA. However, it is worth noticing that the best performing MOEA is not known in advance. Moreover, hyper-heuristic use is faster than searching for the best MOEA by trial and error, with competitive results.

Overall, we conclude that the exchange of information among MOEAs during the search can improve the search ability to achieve better results. When incorporated into a hyper-heuristic framework, the proposed approach is scalable to the number of MOEAs. Besides, after removing the limitation of dividing the search space, it became easy to introduce any type of MOEA and possibly other strategies, such as MOPSOs and MOEDAs. Moreover, the proposed cooperative hyper-heuristic quality generalizes to a wide range of problem instances representing the challenges posed by real-world applications. Finally, we validate those observations by applying the proposed framework to a real-world application with competitive results to the best MOEA. Therefore, the proposed approach is a promising strategy for many-objective optimization in scenarios where the application and fine-tuning of multiple MOEAs individually is not viable. Next, we describe some guidelines for future works.

6.1 FUTURE WORKS

The results achieved in this thesis motivate the investigation of future works. One suggestion is to evaluate the applicability of HH-CO on problems with 2 and 3 objectives. Despite being proposed for many-objective optimization, the HH-CO generality may achieve good results even for problems with two or three objectives. Besides, HH-CO has no parameters to be set, and therefore, its application to another scenario should be straight forward. Moreover, to evaluate the application of HH-CO on discrete problems, one just needs to adjust the pool of MOEAs. Another suggestion is to increase the number of MOEAs in the pool, including the latest MOEAs from the literature, to enhance diversity and achieve even more generality. However, as higher, the number of MOEAs harder to evaluate and select the one to be applied.

We also suggest the study on a minimal effective subset based on a broader set of MOEAs (and other strategies). This can be achieved by the research on offline hyper-heuristics. Another suggestion is the study of different migration typologies. Currently, the computational cost difference compared to an average MOEA is caused by the migration procedure being broadcast. A different topology, where the applied MOEA communicates only to a subset of neighbors, reduces the computational cost. However, there is an open question on what topology to use. One option is to use an adaptive topology that measures the quality of the information exchange of two algorithms and decides what the best topology for the problem at hand is. However, the construction of such a topology is not trivial. It requires the study of many questions, for instance, how to evaluate the quality of the information exchange. Also, future works include using other metrics in the proposed credit assignment scheme. The motivation is that the R2 indicator uses a set of weight vectors that sometimes does not match the distribution of solutions in the true Pareto front. Finally, another way to reduce the computational cost is to implement the migration step in parallel.

REFERENCES

Bader, J. and Zitzler, E. (2011). Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary computation*, 19:45–76.

Bi, X. and Wang, C. (2016). An improved NSGA-III algorithm based on objective space decomposition for many-objective optimization. *Soft Computing*, pages 1–28.

Brockhoff, D., Wagner, T., and Trautmann, H. (2012). On the properties of the r2 indicator. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, GECCO '12, page 465–472, New York, NY, USA. Association for Computing Machinery.

Burke, E. K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., and Qu, R. (2013). Hyper-heuristics: a survey of the state of the art. *Journal of the Operational Research Society*, 64(12):1695–1724.

Burke, E. K., Hyde, M. R., Kendall, G., Ochoa, G., Özcan, E., and Woodward, J. R. (2019). *A Classification of Hyper-Heuristic Approaches: Revisited*, pages 453–477. Springer International Publishing, Cham.

Castro, O. R., Fritsche, G. M., and Pozo, A. (2018). Evaluating selection methods on hyper-heuristic multi-objective particle swarm optimization. *Journal of Heuristics*, 24(4):581–616.

Castro, O. R. and Pozo, A. (2015). Using Hyper-Heuristic to Select Leader and Archiving Methods for Many-Objective Problems. Springer International Publishing.

Castro, O. R., Santana, R., Lozano, J. A., and Pozo, A. (2017). Combining CMA-ES and MOEA/DD for many-objective optimization. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 1451–1458.

Cheng, R., Li, M., Tian, Y., Zhang, X., and Yang, S. (2018). Benchmark Functions for CEC ' 2018 Competition on Evolutionary Many-Objective Optimization. Technical Report January.

Cheng, R., Li, M., Tian, Y., Zhang, X., Yang, S., Jin, Y., and Yao, X. (2017). A benchmark test suite for evolutionary many-objective optimization. *Complex & Intelligent Systems*, 3(1):67–81.

Coello, C. C., Lamont, G. B., and van Veldhuizen, D. A. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Springer.

Das, I. and Dennis, J. E. (1998). Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM Journal on Optimization*, 8(3):631–657.

de Carvalho, V. R. and Sichman, J. S. (2018). Multi-agent election-based hyper-heuristics. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 5779–5780. International Joint Conferences on Artificial Intelligence Organization.

Deb, K. and Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005). *Scalable Test Problems for Evolutionary Multiobjective Optimization*, pages 105–145. Springer London, London.

Demeester, P., Bilgin, B., De Causmaecker, P., and Vanden Berghe, G. (2012). A hyperheuristic approach to examination timetabling problems: benchmarks and a new problem from practice. *Journal of Scheduling*, 15(1):83–103.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30.

Drake, J. H., Kheiri, A., Özcan, E., and Burke, E. K. (2020). Recent Advances in Selection Hyper-heuristics. *European Journal of Operational Research*, 285(2):405–428.

Fan, Z., Li, W., Cai, X., Li, H., Wei, C., Zhang, Q., Deb, K., and Goodman, E. (2020). Difficulty adjustable and scalable constrained multiobjective test problem toolkit. *Evolutionary Computation*, 28(3):339–378.

Figueiredo, E. M. d. N. (2013). Algoritmo baseado em enxame de partículas para otimização de problemas com muitos objetivos. Master's thesis, UFPE, Centro de Informática , Programa de Pós-graduação em Ciência da Computação, Recife, Brasil.

Fleming, P. J., Purshouse, R. C., and Lygoe, R. J. (2005). Many-Objective Optimization: An Engineering Design Perspective. In Coello Coello, Carlos A. and Hernández Aguirre, Arturo and Zitzler, Eckart, editor, *Evolutionary Multi-Criterion Optimization*, pages 14–32, Berlin, Heidelberg. Springer Berlin Heidelberg.

Fritsche, G. and Pozo, A. (2017a). A Distributed Framework for Cooperation of Many-objective Evolutionary Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '17, pages 1804–1811, New York, NY, USA. ACM.

Fritsche, G. and Pozo, A. (2017b). A heterogeneous distributed approach for many-objective optimization. In *2017 Brazilian Conference on Intelligent Systems (BRACIS)*, pages 288–293.

Fritsche, G. M. and Pozo, A. T. R. (2018). A hyper-heuristic collaborative multi-objective evolutionary algorithm. In *7th Brazilian Conference on Intelligent Systems, BRACIS 2018, São Paulo, Brazil, October 22-25, 2018*, pages 354–359. IEEE Computer Society.

Fritsche, G. M. and Pozo, A. T. R. (2019). Cooperative based hyper-heuristic for many-objective optimization. In Auger, A. and Stützle, T., editors, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2019, Prague, Czech Republic, July 13-17, 2019*, pages 550–558. ACM.

Fritsche, G. M. and Pozo, A. T. R. (2020). The analysis of a cooperative hyper-heuristic on a constrained real-world many-objective continuous problem. In *IEEE Congress on Evolutionary Computation, CEC 2020, Glasgow, United Kingdom, July 19-24, 2020*, pages 1–8. IEEE.

Gómez, R. H. and Coello, C. A. C. (2015). Improved metaheuristic based on the r2 indicator for many-objective optimization. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO '15, pages 679–686, New York, NY, USA. ACM.

Gonçalves, R. A., de Almeida, C. P., Lüders, R., and Delgado, M. R. (2018). A new hyper-heuristic based on a contextual multi-armed bandit for many-objective optimization. *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8.

Gonçalves, R. A., Pavelski, L. M., Almeida, C. P., Kuk, J. N., Venske, S. M., and Delgado, M. R. (2017). Adaptive operator selection for many-objective optimization with nsga-iii. In *9th International Conference on Evolutionary Multi-Criterion Optimization - Volume 10173*, EMO 2017, pages 267–281, Berlin, Heidelberg. Springer-Verlag.

Gong, Y.-J., Chen, W.-N., Zhan, Z.-H., Zhang, J., Li, Y., Zhang, Q., and Li, J.-J. (2015). Distributed evolutionary algorithms and their models: A survey of the state-of-the-art. *Applied Soft Computing*, 34:286 – 300.

Guizzo, G., Vergilio, S. R., Pozo, A. T., and Fritsche, G. M. (2017). A multi-objective and evolutionary hyper-heuristic applied to the integration and test order problem. *Applied Soft Computing*, 56:331 – 344.

H. Jain and K. Deb (2014). An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach. *IEEE Transactions on Evolutionary Computation*, 18(4):602–622.

He, Z. and Yen, G. G. (2014). Diversity improvement in decomposition-based multi-objective evolutionary algorithm for many-objective optimization problems. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2409–2414.

Herrero, J. G., Berlanga, A., and Lopez, J. M. M. (2009). Effective Evolutionary Algorithms for Many-Specifications Attainment: Application to Air Traffic Control Tracking Filters. *IEEE Transactions on Evolutionary Computation*, 13(1):151–168.

Huband, S., Barone, L., While, L., and Hingston, P. (2005). A scalable multi-objective test problem toolkit. In Coello Coello, C. A., Hernández Aguirre, A., and Zitzler, E., editors, *Evolutionary Multi-Criterion Optimization*, pages 280–295, Berlin, Heidelberg. Springer Berlin Heidelberg.

Hughes, E. J. (2007). Radar Waveform Optimisation as a Many-Objective Application Benchmark. In Obayashi, Shigeru and Deb, Kalyanmoy and Poloni, Carlo and Hiroyasu, Tomoyuki and Murata, Tadahiko, editor, *Evolutionary Multi-Criterion Optimization*, pages 700–714, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ishibuchi, H., Matsumoto, T., Masuyama, N., and Nojima, Y. (2020). Many-objective problems are not always difficult for pareto dominance-based evolutionary algorithms. In Giacomo, G. D., Catalá, A., Dilkina, B., Milano, M., Barro, S., Bugarín, A., and Lang, J., editors, *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 291–298. IOS Press.

Ishibuchi, H., Setoguchi, Y., Masuda, H., and Nojima, Y. (2017). Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes. *IEEE Transactions on Evolutionary Computation*, 21(2):169–190.

Ishibuchi, H., Tsukamoto, N., and Nojima, Y. (2008). Evolutionary many-objective optimization: A short review. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 2419–2426.

Jaimes, A. L., Oyama, A., and Fujii, K. (2013). Space trajectory design: Analysis of a real-world many-objective optimization problem. In *2013 IEEE Congress on Evolutionary Computation*, pages 2809–2816.

Khan, W. and Zhang, Q. (2010). MOEA/D-DRA with two crossover operators. In *2010 UK Workshop on Computational Intelligence (UKCI)*, pages 1–6.

Li, B., Li, J., Tang, K., and Yao, X. (2015a). Many-objective evolutionary algorithms: A survey. *ACM Comput. Surv.*, 48(1):13:1–13:35.

Li, K., Deb, K., and Zhang, Q. (2015b). Evolutionary multiobjective optimization with hybrid selection principles. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 900–907.

Li, K., Deb, K., Zhang, Q., and Kwong, S. (2015c). An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, 19(5):694–716.

Li, K., Fialho, A., Kwong, S., and Zhang, Q. (2014a). Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 18(1):114–130.

Li, K., Wang, R., Zhang, T., and Ishibuchi, H. (2018a). Evolutionary many-objective optimization: A comparative study of the state-of-the-art. *IEEE Access*, 6:26194–26214.

Li, K., Zhang, Q., Kwong, S., Li, M., and Wang, R. (2014b). Stable matching-based selection in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 18(6):909–923.

Li, M., Yang, S., and Liu, X. (2014c). Shift-based density estimation for pareto-based algorithms in many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 18(3):348–365.

Li, M., Yang, S., and Liu, X. (2016). Pareto or non-pareto: Bi-criterion evolution in multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 20(5):645–665.

Li, W., Özcan, E., and John, R. (2018b). A learning automata based multiobjective hyper-heuristic. *IEEE Transactions on Evolutionary Computation*, pages 1–1.

Liu, H. L., Chen, L., Zhang, Q., and Deb, K. (2017). Adaptively Allocating Search Effort in Challenging Many-Objective Optimization Problems. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1.

Liu, H. L., Gu, F., and Zhang, Q. (2014). Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE Transactions on Evolutionary Computation*, 18(3):450–455.

Liu, Y., Zhu, N., Li, K., Li, M., Zheng, J., and Li, K. (2020). An angle dominance criterion for evolutionary many-objective optimization. *Information Sciences*, 509:376 – 399.

Maashi, M., Kendall, G., and Özcan, E. (2015). Choice function based hyper-heuristics for multi-objective optimization. *Applied Soft Computing*, 28:312 – 326.

Maashi, M., Özcan, E., and Kendall, G. (2014). A multi-objective hyper-heuristic based on choice function. *Expert Systems with Applications*, 41(9):4475 – 4493.

Miettinen, K. (1998). *Nonlinear Multiobjective Optimization*. Springer US.

Musselman, K. and Talavage, J. (1980). A Tradeoff Cut Approach to Multiple Objective Optimization. *Operations Research*, 28(6):1424–1435.

Rodemann, T., Narukawa, K., Fischer, M., and Awada, M. (2015). Many-Objective Optimization of a Hybrid Car Controller. In Mora, Antonio M. and Squillero, Giovanni, editor, *Applications of Evolutionary Computation*, pages 593–603, Cham. Springer International Publishing.

Safi, H. H., Ucan, O. N., and Bayat, O. (2018). On the Real World Applications of Many-objective Evolutionary Algorithms. In *Proceedings of the First International Conference on Data Science, E-learning and Information Systems*, DATA '18, pages 32:1–32:6, New York, NY, USA. ACM.

Srinivas, N. and Deb, K. (1994). Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evol. Comput.*, 2(3):221–248.

Sülflow, A., Drechsler, N., and Drechsler, R. (2007). Robust multi-objective optimization in high dimensional spaces. In Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., and Murata, T., editors, *Evolutionary Multi-Criterion Optimization*, pages 715–726, Berlin, Heidelberg. Springer Berlin Heidelberg.

Sun, Y., Xue, B., Zhang, M., and Yen, G. G. (2019a). A New Two-Stage Evolutionary Algorithm for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation*, 23(5):748–761.

Sun, Y., Yen, G. G., and Yi, Z. (2019b). IGD Indicator-Based Evolutionary Algorithm for Many-Objective Optimization Problems. *IEEE Transactions on Evolutionary Computation*, 23(2):173–187.

Tanabe, R. and Ishibuchi, H. (2020). An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, 89:106078.

The Japanese Society of Evolutionary Computation (2019). The 3rd Evolutionary Computation Competition. http://www.jpnsec.org/files/competition2019/EC-Symposium-2019-Competition-English.html. Last checked on October 2020.

Vázquez-Rodríguez, J. A. and Petrovic, S. (2013). A mixture experiments multi-objective hyper-heuristic. *Journal of the Operational Research Society*, 64(11):1664–1675.

Vrugt, J. A. and Robinson, B. A. (2007). Improved evolutionary optimization from genetically adaptive multimethod search. *Proceedings of the National Academy of Sciences*, 104(3):708–711.

Wan, K., He, C., Camacho, A., Shang, K., Cheng, R., and Ishibuchi, H. (2019). A Hybrid Surrogate-Assisted Evolutionary Algorithm for Computationally Expensive Many-Objective Optimization. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 2018–2025.

Wang, F., Li, Y., Zhang, H., Hu, T., and Shen, X.-L. (2019). An adaptive weight vector guided evolutionary algorithm for preference-based multi-objective optimization. *Swarm and Evolutionary Computation*, 49:220 – 233.

Wang, H., Jiao, L., and Yao, X. (2015a). Two_Arch2: An Improved Two-Archive Algorithm for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation*, 19(4):524–541.

Wang, W., Yang, S., Lin, Q., Zhang, Q., Wong, K., Coello, C. A. C., and Chen, J. (2018). An effective ensemble framework for multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, pages 1–1.

Wang, Z., Zhang, Q., and Li, H. (2015b). Balancing Convergence and Diversity by Using Two Different Reproduction Operators in MOEA/D: Some Preliminary Work. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pages 2849–2854.

Wang, Z. J., Zhan, Z. H., and Zhang, J. (2016). Parallel multi-strategy evolutionary algorithm using massage passing interface for many-objective optimization.

Wickham, H. and Stryjewski, L. (2012). 40 years of boxplots. Technical report, had.co.nz.

Wolpert, D. H. and Macready, W. G. (1997). No Free Lunch Theorems for Optimization. *Trans. Evol. Comp*.

Xiong, M., Xiong, W., and Jian, P. (2019). Visualization of the non-dominated solutions in many-objective optimization. In *2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC)*, pages 188–195.

Xue, Y., Li, M., Shepperd, M., Lauria, S., and Liu, X. (2019). A novel aggregation-based dominance for pareto-based evolutionary algorithms to configure software product lines. *Neurocomputing*, 364:32 – 48.

Yuan, Y., Xu, H., Wang, B., and Yao, X. (2016). A new dominance relation-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 20(1):16–37.

Zhang, Q. and Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731.

Zhang, Q., Liu, W., and Li, H. (2009). The performance of a new version of MOEA/D on CEC09 unconstrained mop test instances. In *2009 IEEE Congress on Evolutionary Computation*, pages 203–208.

Zhang, Y. H., Gong, Y. J., Zhang, J., and b. Ling, Y. (2016). A hybrid evolutionary algorithm with dual populations for many-objective optimization. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 1610–1617.

Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195.

Zitzler, E., Laumanns, M., and Thiele, L. (2001). SPEA2: Improving the strength pareto evolutionary algorithm. Technical report.

APPENDIX A – SCALABILITY ANALYSIS ON THE PROPOSED DISTRIBUTED COOPERATION

This appendix presents some analysis of the scalability of the proposed synchronous distributed cooperation of multiple MOEAs (HeDi). First, we increase the number of MOEAs to three. Moreover, instead of dividing the population size by three, it divided the number of iterations. Also, it was evaluated for the use of a MOEA, which is not decomposition-based: NSGA-II. Along with other conclusions, the results demonstrated that the implementation of NSGA-III was not coherent with the literature results for the same problems. Therefore, we evaluated a different NSGA-III implementation, which presented better results. Finally, we further increased the number of MOEAs to four, with a small population for each one. Based on the conclusions of those experiments, we decided to include the use of hyper-heuristics. Moreover, we decided also to change the benchmark problem set used.

A.1 INCREASE IN THE NUMBER OF MOEAS: DIVIDING THE TOTAL ITERATIONS

In this section, the number of MOEAs involved in the distributed model (HeDi) was increased to three. These are NSGA-III, MOEA/DD, and NSGA-II. The goal was to evaluate the HeDi increasing the number of MOEAs. The experiments were carried out to follow the same methodology presented in Section 5.2.

The results obtained for the hypervolume quality indicator are presented in Table A.1. The best results were obtained by the cooperation or by MOEA/DD being executed alone. In detail: the cooperation achieved the best average of 12 out of the 24 instances assessed by the hypervolume (the hypervolume was not calculated for 15 objectives). From those, at seven problems, the MOEA/DD had no statistically significant difference, leaving 5 in which the cooperation was statistically better than the MOEA/DD. On the other hand, MOEA/DD also obtained the best average hypervolume in 12 instances. The cooperation had no significant difference at just 3. Finally, the cooperation was better or equivalent to the best in 15 out of 24 instances (62.5%).

Analyzing the IGD indicator (Table A.2), the cooperation got the best average in just 8 out of 30 instances. MOEA/DD obtained the best results (21 of 30 problems), with a statistical difference for the others in many cases (16 out of 30). The results obtained in the IGD indicator highlight the performance of MOEA/DD in the problems of the DTLZ family compared to NSGA-III and NSGA-II. This observation suggests that the difference in the performance of the considered MOEAs may have affected the cooperation. So, in general, for the IGD, the MOEA/DD executed alone has better results.

The critical difference plot (Figures A.1 and A.2) supports the observations obtained by Tables A.1 and A.2. In the hypervolume, HeDi and MOEA/DD achieved the same average ranking. Being statistically different from NSGA-II and NSGA-III (which were statistically equivalent to each other). For IGD, the conclusions are the same. The difference is that the MOEA/DD obtained an average ranking higher than the cooperation. Of these results, we highlight that the NSGA-II achieved the best average ranking than NSGA-III. This behavior is inconsistent with the quality of NSGA-III reported in the literature. In the following appendix, the NSGA-III implementation used so far is compared with another available online.

Table A.1: Average (and standard deviation) for the Hypervolume indicator. The best value for each instance is shown in bold; values statistically equivalent to the best are presented on a gray background, with 95% significance.

Obj.	problem	HeDi	NSGA-III	NSGA-II	MOEA/DD
3	DTLZ1	**9,74E-1(5,24E-4)**	8,69E-1(9,83E-2)	9,68E-1(3,79E-3)	9,73E-1(1,21E-4)
	DTLZ2	**7,42E0(1,22E-3)**	6,5E0(3,95E-1)	7,35E0(2,36E-2)	7,41E0(9,39E-5)
	DTLZ3	**7,41E0(6,81E-3)**	5,7E0(7,85E-1)	7,36E0(2,39E-2)	7,41E0(5,43E-3)
	DTLZ4	**7,43E0(8,51E-4)**	6,85E0(4,49E-1)	7,03E0(1,04E0)	7,41E0(7,89E-6)
	WFG6	7,17E1(5,79E-1)	5,54E1(4,59E0)	6,97E1(6,68E-1)	**7,22E1(5,39E-1)**
	WFG7	**7,54E1(2,28E-1)**	4,11E1(5,81E0)	7,31E1(6,35E-1)	7,53E1(2,1E-1)
5	DTLZ1	9,99E-1(1,49E-4)	9,64E-1(2,74E-2)	0E0(0E0)	**9,99E-1(3,07E-6)**
	DTLZ2	**3,17E1(2,43E-3)**	2,93E1(1,22E0)	3,05E1(3,48E-1)	3,17E1(2,38E-4)
	DTLZ3	3,17E1(2,55E-2)	2,54E1(1,01E1)	0E0(0E0)	**3,17E1(1,78E-3)**
	DTLZ4	**3,17E1(2,79E-3)**	3,05E1(5,85E-1)	3,14E1(6,51E-2)	3,17E1(3,89E-6)
	WFG6	8,51E3(9,25E1)	4,28E3(5,16E2)	7,39E3(1,7E2)	**8,59E3(4,28E1)**
	WFG7	8,82E3(5,95E1)	3,79E3(5,33E2)	7,53E3(1,97E2)	**8,98E3(2,56E1)**
8	DTLZ1	9,99E-1(6,79E-4)	6,48E-1(2,92E-1)	0E0(0E0)	**1E0(1,71E-5)**
	DTLZ2	2,56E2(1,75E-2)	2,11E2(2,16E1)	1,66E1(1,23E1)	**2,56E2(2,34E-3)**
	DTLZ3	2,49E2(2,12E1)	0E0(0E0)	0E0(0E0)	**2,56E2(4,52E-3)**
	DTLZ4	**2,56E2(3E-3)**	2,17E2(1,54E1)	6,72E1(4,16E1)	2,56E2(1,28E-5)
	WFG6	**2,96E7(5,46E5)**	6,03E6(1,08E6)	1,89E7(9,54E5)	2,94E7(3,54E5)
	WFG7	3,08E7(2,37E5)	6,16E6(7,86E5)	1,86E7(8,18E5)	**3,12E7(1,72E5)**
10	DTLZ1	1E0(7,86E-5)	7,42E-1(2,99E-1)	0E0(0E0)	**1E0(2,81E-6)**
	DTLZ2	1,02E3(1,59E-2)	8,83E2(4,39E1)	5,27E1(4,36E1)	**1,02E3(7,29E-4)**
	DTLZ3	1,02E3(3,93E-2)	0E0(0E0)	0E0(0E0)	**1,02E3(7,65E-4)**
	DTLZ4	**1,02E3(4,32E-3)**	8,4E2(5,13E1)	3,25E2(1,48E2)	1,02E3(1,08E-5)
	WFG6	**1,21E10(2,16E8)**	2,3E9(3,98E8)	7,34E9(3,01E8)	1,17E10(1,29E8)
	WFG7	**1,27E10(4,59E7)**	2,54E9(4,16E8)	7,17E9(2,82E8)	1,26E10(3,62E7)

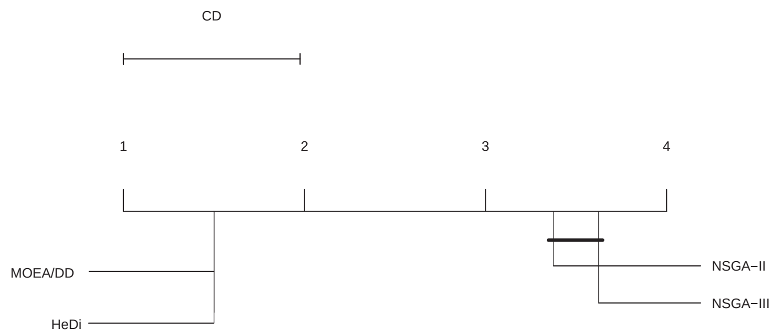


Figure A.1: Critical difference plot for the hypervolume indicator. The algorithms connected by a bold horizontal line are considered statistically equivalent with 95% of significance.

A.2 EVALUATING DIFFERENT IMPLEMENTATIONS OF NSGA-III

Given the bad results presented by the NSGA-III in previous experiments, it became necessary to validate its implementation. For this matter, it was compared to another implementation of NSGA-III[1]. That was the implementation used by Yuan et al. (2016) and by Ishibuchi et al. (2017). For this comparison, we used the same methodology from previous experiments. Assessed by both hypervolume and IGD (see Tables A.3 and A.4). In both cases, for all evaluated problem instances, the competitor implementation achieved better results. It demonstrated a statistically significant difference, compared to the version from jMetal used so far in this research[2]. Based

[1]The NSGA-III implementation used for comparison is available at `https://github.com/yyxhdy/ManyEAs`

[2]We reported those results to the jMetal repository issue section.

Table A.2: Average (and standard deviation) for the IGD indicator. The best value for each instance is shown in bold; values statistically equivalent to the best are presented on a gray background, with 95% significance.

Obj.	problem	HeDi	NSGA-III	NSGA-II	MOEA/DD
3	DTLZ1	5,6E-4(2,84E-4)	2,82E-2(1,76E-2)	7,36E-3(6,62E-4)	**1,74E-4(9,62E-5)**
	DTLZ2	5,4E-4(2,08E-4)	2,01E-2(6,17E-3)	9,1E-3(3,91E-4)	**1,21E-4(3,2E-5)**
	DTLZ3	5,86E-4(3,03E-4)	4,53E-2(1,87E-2)	9,32E-3(6,7E-4)	**4,18E-4(3,36E-4)**
	DTLZ4	1,08E-4(3,19E-5)	3,82E-2(3,07E-2)	1,89E-2(3,06E-2)	**3,43E-5(4,98E-5)**
	WFG6	4,51E-3(6,11E-4)	2,06E-2(5,16E-3)	1,01E-2(5,72E-4)	**3,49E-3(4,29E-4)**
	WFG7	**2,66E-3(1,67E-4)**	4,11E-2(1,11E-2)	9,07E-3(7,44E-4)	2,68E-3(3,29E-4)
5	DTLZ1	8,96E-4(2,78E-4)	1,96E-2(6,06E-3)	8,91E-1(7,95E-1)	**7,51E-5(6,01E-5)**
	DTLZ2	7,38E-4(7,86E-5)	2,13E-2(3,7E-3)	2,62E-2(2,43E-3)	**1,02E-4(9,27E-6)**
	DTLZ3	2,35E-3(1,57E-3)	4,79E-2(5,34E-2)	7,53E0(3,9E0)	**1,26E-4(7,91E-5)**
	DTLZ4	1,23E-4(3,7E-5)	3,58E-2(1,03E-2)	1,82E-2(7,64E-4)	**1,05E-5(2,21E-6)**
	WFG6	**8,25E-3(1,8E-4)**	2,84E-2(4,41E-3)	1,51E-2(6,11E-4)	8,66E-3(4,18E-5)
	WFG7	**1,32E-2(7,09E-4)**	3,69E-2(8,61E-3)	1,75E-2(6,79E-4)	1,91E-2(1,06E-4)
8	DTLZ1	4,13E-3(7,31E-4)	8,01E-2(3,82E-2)	3,61E0(1,26E0)	**6,35E-4(3,78E-4)**
	DTLZ2	2,21E-3(2,04E-4)	5,85E-2(6,91E-3)	1,48E-1(1,41E-2)	**4,07E-4(9,58E-5)**
	DTLZ3	1,63E-2(2,55E-2)	6,94E0(2,88E0)	1,75E1(7,07E0)	**6,56E-4(2,13E-4)**
	DTLZ4	8,59E-4(1,75E-4)	8,48E-2(6,05E-3)	1,28E-1(1,31E-2)	**1,35E-4(7,23E-5)**
	WFG6	**3,04E-2(8,17E-4)**	6,29E-2(5,82E-3)	3,66E-2(1,45E-3)	3,59E-2(1,03E-3)
	WFG7	**4,57E-2(2,51E-3)**	7,49E-2(7E-3)	5,71E-2(3,03E-3)	6,06E-2(1,06E-3)
10	DTLZ1	2,27E-3(4,45E-4)	5,65E-2(3,15E-2)	2,78E0(1,31E0)	**4,1E-4(1,22E-4)**
	DTLZ2	2E-3(1,1E-4)	4,49E-2(4,32E-3)	1,18E-1(1,47E-2)	**6,65E-4(1,07E-4)**
	DTLZ3	3,17E-3(5,91E-4)	4,81E0(2,63E0)	1,23E1(2,88E0)	**7,11E-4(8,32E-5)**
	DTLZ4	1,08E-3(1,05E-4)	6,96E-2(2,85E-3)	9,77E-2(9,99E-3)	**3,72E-4(5,31E-5)**
	WFG6	**3,23E-2(1,21E-3)**	4,76E-2(3,74E-3)	3,51E-2(1,14E-3)	4,66E-2(1,16E-3)
	WFG7	**5,3E-2(1,86E-3)**	7,54E-2(8,79E-3)	7,45E-2(4,08E-3)	6,21E-2(3,47E-4)
15	DTLZ1	6,36E-3(1,98E-3)	9,5E-2(3,64E-2)	4,75E0(1,73E0)	**1,69E-3(7,55E-4)**
	DTLZ2	3,43E-3(2,2E-4)	9,82E-2(3,1E-3)	1,78E-1(1,65E-2)	**2,21E-3(2,2E-4)**
	DTLZ3	4,46E-3(4,66E-4)	2,7E0(3,72E0)	1,33E1(3,75E0)	**2,39E-3(1,84E-4)**
	DTLZ4	3,18E-3(2,86E-4)	1,09E-1(3,5E-3)	1,14E-1(5,66E-3)	**1,87E-3(4,3E-4)**
	WFG6	1,13E-1(9,87E-3)	1,21E-1(5,7E-3)	**1,12E-1(7,15E-3)**	1,94E-1(2,48E-2)
	WFG7	**1,42E-1(8,55E-3)**	2,03E-1(4,99E-2)	1,42E-1(8,48E-3)	1,91E-1(6,08E-2)

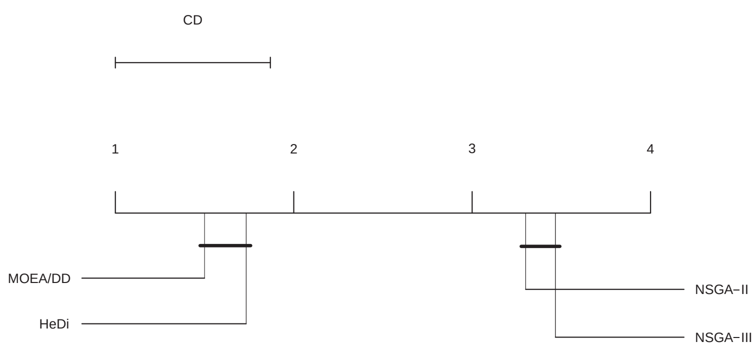


Figure A.2: Critical difference plot for the IGD indicator. The algorithms connected by a bold horizontal line are considered statistically equivalent with 95% of significance.

on those results, the implementation of NSGA-III from Yuan et al. (2016) is used in all further experiments with NSGA-III. Those observations do not affect the conclusions reported for the cooperative frameworks evaluated so far. However, the previous observations about NSGA-III should be considered valid only for the jMetal implementation.

Table A.3: Average (and standard deviation) for the Hypervolume indicator. The best value for each instance is shown in bold; values statistically equivalent to the best are presented on a gray background, with 95% significance.

Obj.	problem	NSGA-III from jMetal	NSGA-III from ManyEAs
3	DTLZ1	8,69E-1(9,83E-2)	**9,73E-1(5,42E-4)**
	DTLZ2	6,5E0(3,95E-1)	**7,41E0(6,9E-4)**
	DTLZ3	5,7E0(7,85E-1)	**7,41E0(3,95E-3)**
	DTLZ4	6,85E0(4,49E-1)	**7,07E0(1,05E0)**
	WFG6	5,54E1(4,59E0)	**7,23E1(4,91E-1)**
	WFG7	4,11E1(5,81E0)	**7,56E1(1,69E-1)**
5	DTLZ1	9,64E-1(2,74E-2)	**9,99E-1(1,15E-4)**
	DTLZ2	2,93E1(1,22E0)	**3,17E1(5,14E-4)**
	DTLZ3	2,54E1(1,01E1)	**3,17E1(3,4E-3)**
	DTLZ4	3,05E1(5,85E-1)	**3,17E1(3,18E-4)**
	WFG6	4,28E3(5,16E2)	**8,63E3(4,72E1)**
	WFG7	3,79E3(5,33E2)	**9,06E3(2,02E1)**
8	DTLZ1	6,48E-1(2,92E-1)	**1E0(5,51E-6)**
	DTLZ2	2,11E2(2,16E1)	**2,56E2(1,38E-3)**
	DTLZ3	0E0(0E0)	**2,56E2(4,57E-2)**
	DTLZ4	2,17E2(1,54E1)	**2,56E2(2,16E-4)**
	WFG6	6,03E6(1,08E6)	**3,08E7(2,74E5)**
	WFG7	6,16E6(7,86E5)	**3,22E7(5,77E4)**
10	DTLZ1	7,42E-1(2,99E-1)	**1E0(5,31E-7)**
	DTLZ2	8,83E2(4,39E1)	**1,02E3(1,07E-3)**
	DTLZ3	0E0(0E0)	**1,02E3(4,63E-3)**
	DTLZ4	8,4E2(5,13E1)	**1,02E3(1,18E-4)**
	WFG6	2,3E9(3,98E8)	**1,26E10(9,83E7)**
	WFG7	2,54E9(4,16E8)	**1,33E10(1,64E7)**

A.3 INCREASE IN THE NUMBER OF MOEAS: DIVIDING THE POPULATION SIZE

We proceeded with evaluating the proposed distributed cooperation — after the validation and change of NSGA-III implementation. In this section, we assess a new approach to improve scalability and increase the number of MOEAs. In this approach, we divide the default population size by four MOEAs. It is different from the strategy used in Section A.1. The MOEAs used in this experiment are NSGA-II and SPEA2 (from jMetal), ThetaDEA, and NSGA-III (from the ManyEAs repository). We used the same methodology from previous experiments.

The proposed approach for cooperation used in this section follows. First, all MOEAs are executed and generate offspring from their subpopulation. Then, we combine all the generated offspring. Finally, the migration step sends this set to the environmental selection step of each MOEA. Moreover, different from the first experiments with distributed MOEAs, this approach did not decompose the objective space for different directions for each MOEA. The NSGA-II and SPEA2 are not decomposition-based. Both ThetaDEA and NSGA-III allow the use of more weight vectors than solutions. Therefore, they used the full set of weight vectors.

Finally, we analyzed this approach results using the hypervolume quality indicator. The hypervolume analysis revealed that NSGA-III and ThetaDEA, applied standalone, achieved better results than the cooperation of MOEAs (Table A.5). The critical difference plot demonstrated that the ThetaDEA and NSGA-III achieved the best average ranking, without a significant difference to each other (Figure A.3). However, they performed better results, with a significant difference, compared to the cooperation. The cooperation achieved a better average ranking than the Pareto based MOEAs, without a significant difference to NSGA-II.

The algorithms like ThetaDEA and NSGA-III are known as better options than SPEA2 and NSGA-II for the DTLZ and WFG problem sets (Ishibuchi et al., 2017). The median results achieved by the distributed cooperation may be caused by the significant difference between the quality of the better and worse MOEAs for this problem set. Moreover, the population size

Table A.4: Average (and standard deviation) for the IGD indicator. The best value for each instance is shown in bold; values statistically equivalent to the best are presented on a gray background, with 95% significance.

Obj.	problem	NSGA-III from jMetal	NSGA-III from ManyEAs
3	DTLZ1	2,82E-2(1,76E-2)	**1,45E-3(1,42E-3)**
	DTLZ2	2,01E-2(6,17E-3)	**4,15E-4(2,28E-4)**
	DTLZ3	4,53E-2(1,87E-2)	**9,06E-4(1,5E-3)**
	DTLZ4	3,82E-2(3,07E-2)	**1,11E-2(3,32E-2)**
	WFG6	2,06E-2(5,16E-3)	**6,3E-3(1,89E-4)**
	WFG7	4,11E-2(1,11E-2)	**5,91E-3(3,72E-5)**
5	DTLZ1	1,96E-2(6,06E-3)	**2,08E-3(3,05E-3)**
	DTLZ2	2,13E-2(3,7E-3)	**5,95E-4(9,01E-5)**
	DTLZ3	4,79E-2(5,34E-2)	**1,09E-3(1,02E-3)**
	DTLZ4	3,58E-2(1,03E-2)	**1,91E-4(9,01E-5)**
	WFG6	2,84E-2(4,41E-3)	**1,11E-2(4,63E-5)**
	WFG7	3,69E-2(8,61E-3)	**1,47E-2(4,33E-5)**
8	DTLZ1	8,01E-2(3,82E-2)	**1,64E-3(1,28E-3)**
	DTLZ2	5,85E-2(6,91E-3)	**1,75E-3(1,6E-4)**
	DTLZ3	6,94E0(2,88E0)	**5,74E-3(5,8E-3)**
	DTLZ4	8,48E-2(6,05E-3)	**8,32E-4(1,93E-4)**
	WFG6	6,29E-2(5,82E-3)	**3,44E-2(5,03E-4)**
	WFG7	7,49E-2(7E-3)	**4,48E-2(5,05E-4)**
10	DTLZ1	5,65E-2(3,15E-2)	**1,16E-3(8,76E-4)**
	DTLZ2	4,49E-2(4,32E-3)	**1,42E-3(1,21E-4)**
	DTLZ3	4,81E0(2,63E0)	**1,67E-3(4,9E-4)**
	DTLZ4	6,96E-2(2,85E-3)	**6,89E-4(8,8E-5)**
	WFG6	4,76E-2(3,74E-3)	**3,5E-2(1,87E-4)**
	WFG7	7,54E-2(8,79E-3)	**4,23E-2(1,16E-3)**
15	DTLZ1	9,5E-2(3,64E-2)	**2,24E-3(1,63E-3)**
	DTLZ2	9,82E-2(3,1E-3)	**3,38E-3(3,71E-4)**
	DTLZ3	2,7E0(3,72E0)	**1,09E-2(1,9E-2)**
	DTLZ4	1,09E-1(3,5E-3)	**1,9E-3(3,06E-4)**
	WFG6	1,21E-1(5,7E-3)	**1,14E-1(6,66E-3)**
	WFG7	2,03E-1(4,99E-2)	**1,11E-1(7,24E-3)**

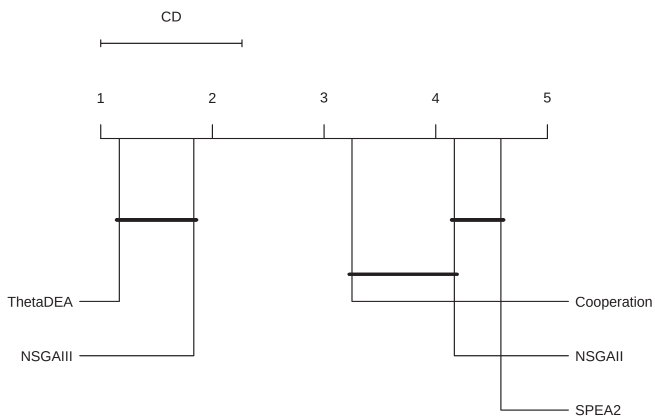


Figure A.3: Critical difference plot for hypervolume quality indicator. A bold horizontal line connects the algorithms without significant statistical difference, with 95% significant.

division makes each MOEA have a tiny population, which causes difficulties for many-objective optimization. For further experiments, we changed the benchmark set. Also, we make use of hyper-heuristics to the cooperation of MOEAs. In this way, it is possible to execute one MOEA at a time without dividing the number of iterations or the population size. Besides, the migration step allows exchanging information from the executed MOEAs and the others.

Table A.5: Average (and standard deviation) for the Hypervolume indicator. The best value for each instance is shown in bold; values statistically equivalent to the best are presented on a gray background, with 95% significance.

Obj.	problem	Cooperation	ThetaDEA	NSGA-III	NSGA-II	SPEA2
3	DTLZ1	9,68E-1(1,41E-3)	**9,73E-1(2,25E-4)**	9,73E-1(2,89E-4)	9,58E-1(4,97E-2)	9,68E-1(1,83E-3)
	DTLZ2	7,36E0(1,54E-2)	**7,41E0(1,93E-4)**	7,41E0(7,1E-4)	7,35E0(1,86E-2)	7,29E0(2,94E-2)
	DTLZ3	7,35E0(1,35E-2)	**7,41E0(5,84E-3)**	7,4E0(7,67E-3)	7,37E0(1,71E-2)	7,27E0(4,71E-2)
	DTLZ4	6,86E0(1,23E0)	7,36E0(2,34E-1)	**7,41E0(3,53E-4)**	7,2E0(7,53E-1)	6,99E0(8,01E-1)
	WFG6	7,05E1(6,77E-1)	**7,24E1(5E-1)**	7,22E1(4,81E-1)	6,97E1(6,59E-1)	6,58E1(8,21E-1)
	WFG7	7,4E1(4,05E-1)	**7,58E1(1,3E-1)**	7,56E1(1,44E-1)	7,33E1(4,34E-1)	6,75E1(1,13E0)
5	DTLZ1	9,92E-1(4,04E-3)	**9,99E-1(7,34E-6)**	9,99E-1(1,98E-5)	0E0(0E0)	0E0(0E0)
	DTLZ2	3,14E1(2,74E-2)	**3,17E1(3,4E-4)**	3,17E1(8,13E-4)	3,03E1(3,32E-1)	3,12E1(7,28E-2)
	DTLZ3	3,06E1(5,09E-1)	**3,17E1(2,6E-3)**	3,17E1(5,8E-3)	0E0(0E0)	0E0(0E0)
	DTLZ4	3,16E1(1,51E-2)	**3,17E1(1,74E-5)**	3,17E1(2,52E-4)	3,14E1(4,95E-2)	3,1E1(1,26E0)
	WFG6	8,15E3(1,89E2)	**8,66E3(6,31E1)**	8,63E3(4,98E1)	7,39E3(1,7E2)	6,25E3(2E2)
	WFG7	8,49E3(1,3E2)	**9,1E3(8,07E0)**	9,06E3(1,86E1)	7,53E3(1,97E2)	6,02E3(1,23E2)
8	DTLZ1	9,78E-1(5,08E-2)	1E0(4,05E-6)	**1E0(3,37E-6)**	0E0(0E0)	0E0(0E0)
	DTLZ2	2,23E2(3,08E1)	**2,56E2(7,75E-4)**	2,56E2(1,88E-3)	1,74E1(2,38E1)	2,37E1(5,98E0)
	DTLZ3	0E0(0E0)	**2,56E2(5,04E-3)**	2,56E2(1,93E-2)	0E0(0E0)	0E0(0E0)
	DTLZ4	2,53E2(1,98E0)	**2,56E2(8,07E-5)**	2,56E2(1,85E-4)	6,72E1(4,16E1)	1,63E1(4,99E0)
	WFG6	2,52E7(9,15E5)	**3,08E7(1,86E5)**	3,08E7(3,41E5)	1,89E7(9,54E5)	1,47E7(9,6E5)
	WFG7	2,68E7(8,83E5)	**3,23E7(5,82E4)**	3,22E7(6,62E4)	1,86E7(8,18E5)	1,45E7(8,08E5)
10	DTLZ1	6,44E-1(3,62E-1)	1E0(1,2E-6)	**1E0(3,3E-7)**	0E0(0E0)	0E0(0E0)
	DTLZ2	8,1E2(1,17E2)	**1,02E3(8,48E-4)**	1,02E3(9,66E-4)	6,09E1(5,09E1)	7,82E1(1,33E1)
	DTLZ3	0E0(0E0)	**1,02E3(1,2E-3)**	1,02E3(5,76E-3)	0E0(0E0)	0E0(0E0)
	DTLZ4	1,01E3(8,2E0)	**1,02E3(4,02E-5)**	1,02E3(1,67E-4)	3,25E2(1,48E2)	6,59E1(9,18E0)
	WFG6	9,93E9(2,94E8)	1,27E10(8,03E7)	**1,27E10(6,97E7)**	7,34E9(3,01E8)	5,47E9(3,05E8)
	WFG7	1E10(3,37E8)	**1,33E10(1,12E7)**	1,33E10(1,67E7)	7,17E9(2,82E8)	5,99E9(3,52E8)

APPENDIX B – PROOFS OF CONCEPT FOR THE USE OF HYPER-HEURISTICS

In this section, we demonstrate proofs of concepts for the use of hyper-heuristics. First, we aim to show the presence of the no free lunch theorem on many-objective problems. We configured experiments manually, setting the probabilities for each MOEA on a hyper-heuristic framework. The goal was to assess, using previous knowledge, that applying the best performing MOEAs more often could contribute to the search. Therefore, we validate the need for the use of hyper-heuristics.

B.1 PROOF OF CONCEPT I: PROBABILITIES MANUALLY SET FOR EACH PROBLEM

In this section, we aim to demonstrate the no free lunch theorem and evaluate hyper-heuristic use capabilities. We show that the best MOEA varies for different problem instances. Also, we assess if the adequate weight of the participation of each MOEA is capable of achieving results comparable to the ones performed by the best MOEAs for the problem. For this experiment, we used seven MOEAs: MOMBI2, ThetaDEA, NSGA-III, MOEA/DD, SPEA2, NSGA-II, and MOEA/D. We selected WFG1 and MinusWFG1 problems, from Ishibuchi et al. (2017), as the best performing MOEAs widely vary for those two different problems. Table B.1 demonstrates the results for the hypervolume quality indicator. It is possible to observe, for instance, that the MOMBI2, followed by MOEA/DD, was the best performing MOEAs for WFG1, with three objectives, being MOEA/D the worst-performing MOEA. On MinusWFG1, with three objectives, the MOEA/D was the best, while MOMBI2 and MOEA/DD were the worst results.

In this preliminary analysis, we built a hyper-heuristic framework with the following characteristics. Every iteration, a MOEA from the pool is selected and applied. The MOEA will execute for one iteration and shares its population with the other MOEAs. We aim at validating the cooperation of multiple MOEAs, guided by hyper-heuristics. The HH weighs the participation of each MOEA depending on its quality. The quality of each one of the seven MOEAs applied is known for this problem. Based on that information, we created a roulette wheel of probabilities for each problem instance. This roulette is used during the search to decide which MOEA is going to be applied. The probabilities were set from $\frac{1}{127}$ for the worst-performing MOEA for that problem to $\frac{64}{127}$ for the best performing MOEA. The MOEAs were sorted, and each subsequent MOEA has double probability than the previous one.

Table B.1: MOEAs ranking for WFG1 and MinusWFG1 with 3 objectives, ranked by hypervolume

WFG1			**MinusWFG1**		
Ranking	**MOEA**	**Prob.** $\times 127$	**Ranking**	**MOEA**	**Prob.** $\times 127$
1	MOMBI2	64	1	MOEA/D	64
2	MOEA/DD	32	2	NSGA-II	32
3	ThetaDEA	16	3	NSGA-III	16
4	NSGA-II	8	4	ThetaDEA	8
5	NSGA-III	4	5	SPEA2	4
6	SPEA2	2	6	MOMBI2	2
7	MOEA/D	1	7	MOEA/DD	1

For both problems (Table B.2), the cooperation achieved the best average hypervolume. However, without a statistically significant difference to the best MOEAs: MOMBI-II, MOEA/DD and ThetaDEA for WFG1, and MOEA/D and NSGA-II for MinusWFG1.

Table B.2: Hypervolume average for the cooperation and MOEAs applied standalone. The best value is presented in boldface, and the values without statistical significant difference to the best are presented with gray background (95% significance).

Obj.	problem	Cooperation	MOEA/D	MOMBI2	MOEA/DD	NSGA-III	ThetaDEA	NSGA-II	SPEA2
3	WFG1	**62.9059**	42.5995	59.7514	57.7372	53.5998	57.5277	53.7452	42.8202
	MinusWFG1	**28.7908**	27.3218	8.21170	7.11622	14.8303	12.0737	18.9469	10.7552

B.2 PROOF OF CONCEPT II: PROBABILITIES BY PROBLEM FAMILY

Unlike previous analysis, we set the probabilities once and used for all problem instances in this section. Table B.3 presents the probabilities. To learn the overall ranking of the evaluated MOEAs, they were evaluated in a set of nine problem instances. We aim to validate the need to learn what MOEAs to apply more often for different problem instances.

Table B.3: MOEAs ranking for WFG benchmark with 3, 5, 8 and 10 objectives (evaluated by average hypervolume).

Ranking	**MOEA**	**Prob** $\times 127$
1	Theta-DEA	64
2	NSGA-III	32
3	MOMBI2	16
4	MOEADD	8
5	NSGA-II	4
6	MOEAD	2
7	SPEA2	1

According to Ishibuchi et al. (2017), the MOEAs performance depends on the Pareto front shape. The WFG4 to WFG9 problems present the same Pareto front shape. Therefore, the ranking of MOEAs is similar to these problems. For instance, ThetaDEA is the best MOEA evaluated for these problems, while SPEA2 was the worst. For the WFG1 to WFG3, the Pareto front shape differs. Therefore, it also varies the ranking of MOEAs for those problems. Those observations are coherent to the results presented in the literature (Ishibuchi et al., 2017).

We built a roulette wheel with higher probabilities for the MOEAs with better performance at WFG4 to WFG9 problems for this analysis. The results observed demonstrate that the cooperation achieved better results for most problem instances. Except for WFG1 with 5 and 8 objectives, the cooperation achieved results as good or better than the best performing MOEA. The bad results presented for WFG1 instances demonstrate that the participation of each MOEA must be learned for different problem instances. In the analysis from proof of concepts I and II, we observed that the performance of MOEAs varies for different problem instances. Therefore, it is not possible to configure a predefined balance for the participation of the MOEAs in the search. Thus, in the next experiments, we propose a hyper-heuristic framework, including methods to evaluate the MOEAs and select the one to be applied during the search. The results achieved by the proofs of concept motivated the use of hyper-heuristics used in Section 4.4.

Table B.4: Average hypervolume for the coopeartion and the MOEAs applied standalone. The best value presented for each problem is highlighted with bold face. The values without statistical significant difference (with 95% significance) to the best are highlighted with gray background.

Obj.	problem	Cooperation	MOEA/DD	NSGA-III	ThetaDEA	NSGA-II	SPEA2	MOMBI2	MOEA/D
	WFG1	60.3606	56.4830	54.2592	57.4383	53.2943	43.8565	**60.7434**	42.5854
	WFG2	**99.4352**	90.7819	90.0472	96.2105	91.3203	93.7578	89.6473	93.4973
	WFG3	**74.5349**	69.8021	71.9752	72.7250	73.6184	70.3799	72.2889	68.0672
	WFG4	**75.3787**	74.7260	74.8198	74.9468	72.0319	66.8436	74.6585	66.0525
3	WFG5	**72.6418**	71.4702	71.8712	71.8670	69.9519	65.8748	71.6922	68.1691
	WFG6	**72.8407**	72.2475	72.2262	72.5208	69.7512	65.3064	72.3528	67.3536
	WFG7	**76.4627**	75.3634	75.6654	75.8077	73.3623	67.8324	75.5255	68.3960
	WFG8	**69.3049**	68.2489	68.2850	68.3518	65.3155	59.4223	67.8815	56.5664
	WFG9	**69.0141**	68.8862	68.4554	68.4569	65.8706	64.2345	67.7054	64.2432
	WFG1	6168.57	6763.78	5929.32	7268.16	5041.39	3829.07	**7936.69**	5600.27
	WFG2	10256.9	10074.6	10064.6	9976.01	**10269.1**	9875.54	9729.91	9917.50
	WFG3	**7390.23**	6453.06	6798.10	6955.41	7166.60	5274.62	6939.74	6101.73
	WFG4	**8947.04**	8901.93	8818.41	8878.68	7557.82	6766.02	8922.26	6591.53
5	WFG5	**8693.02**	8433.87	8586.85	8601.72	7430.31	6674.37	8552.83	6973.48
	WFG6	**8679.87**	8582.93	8645.22	8657.77	7387.38	6261.91	8650.21	7265.01
	WFG7	**9186.58**	8976.47	9060.15	9098.01	7479.54	6062.77	9050.91	7405.02
	WFG8	**8211.14**	7999.07	7955.19	8010.46	6524.49	5480.01	7632.18	4619.41
	WFG9	**7934.77**	7803.92	7705.91	7809.64	6761.60	6304.84	7682.97	6276.29
	WFG1	2.54955e+07	2.50014e+07	2.71509e+07	2.98614e+07	1.50820e+07	1.05066e+07	**3.02974e+07**	2.57562e+07
	WFG2	3.40119e+07	3.24991e+07	3.15724e+07	3.03619e+07	**3.42511e+07**	3.06606e+07	3.06318e+07	3.32591e+07
	WFG3	**2.33663e+07**	1.85675e+07	2.05292e+07	1.68018e+07	2.27804e+07	1.13171e+07	3.68399e+06	1.95878e+07
	WFG4	**3.16973e+07**	3.04079e+07	3.13976e+07	3.13874e+07	2.02148e+07	1.63987e+07	2.92520e+07	1.87389e+07
8	WFG5	**3.07222e+07**	2.81012e+07	3.04527e+07	3.04413e+07	1.79733e+07	1.57997e+07	2.94477e+07	1.93660e+07
	WFG6	**3.08522e+07**	2.93829e+07	3.08008e+07	3.07762e+07	1.93966e+07	1.43942e+07	3.05437e+07	2.13614e+07
	WFG7	**3.25984e+07**	3.12131e+07	3.22389e+07	3.22981e+07	1.82045e+07	1.45346e+07	3.14236e+07	2.05602e+07
	WFG8	**2.91051e+07**	2.73032e+07	2.71319e+07	2.71940e+07	1.85337e+07	1.22056e+07	2.50056e+07	1.40483e+07
	WFG9	2.64492e+07	2.40908e+07	2.60783e+07	**2.66808e+07**	1.53884e+07	1.40111e+07	2.48146e+07	1.76239e+07
	WFG1	1.29803e+10	1.14905e+10	1.18121e+10	1.23742e+10	7.08265e+09	3.91531e+09	1.26580e+10	**1.30579e+10**
	WFG2	1.36632e+10	1.31043e+10	1.26236e+10	1.23371e+10	**1.37075e+10**	1.21469e+10	1.24930e+10	1.36884e+10
	WFG3	**9.52021e+09**	7.11934e+09	7.52058e+09	8.17914e+09	9.20566e+09	4.10326e+09	5.65469e+09	8.61037e+09
	WFG4	**1.30179e+10**	1.22769e+10	1.29986e+10	1.30054e+10	7.34597e+09	6.85962e+09	1.18402e+10	8.58168e+09
10	WFG5	**1.25289e+10**	1.13058e+10	1.24955e+10	1.24940e+10	6.80965e+09	6.29633e+09	1.21090e+10	7.47139e+09
	WFG6	1.26353e+10	1.17819e+10	1.26511e+10	**1.26853e+10**	7.31233e+09	5.53763e+09	1.25802e+10	8.72438e+09
	WFG7	**1.33606e+10**	1.26525e+10	1.32877e+10	1.33044e+10	7.23793e+09	6.01926e+09	1.30001e+10	8.02948e+09
	WFG8	**1.23091e+10**	1.11539e+10	1.14232e+10	1.15302e+10	7.49575e+09	4.40547e+09	1.04800e+10	5.89671e+09
	WFG9	1.07942e+10	9.14865e+09	1.04464e+10	**1.09576e+10**	6.31494e+09	6.30443e+09	1.00769e+10	6.66272e+09

APPENDIX C – A COMPARATIVE STUDY OF NINE STATE-OF-THE-ART MOEAS

This appendix presents a comparative study of the nine MOEAs that compose the heuristics pool of the proposed HH-CO (see Section 4.5). For this study, we make use of the same benchmark and methodology from Section 5.4. This study demonstrates the diversity of MOEAs used in this work. Besides, it confirms the No-Free-Lunch theorem in many-objective optimization. For instance, each one of the nine algorithms was the best for at least one problem instance.

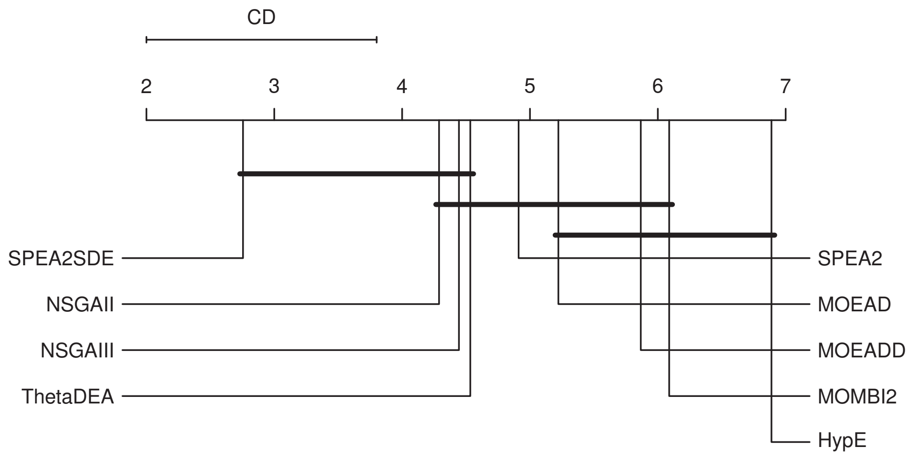


Figure C.1: Critical difference plot for the IGD indicator. The algorithms connected by a bold horizontal line are considered statistically equivalent with 95% of significance.

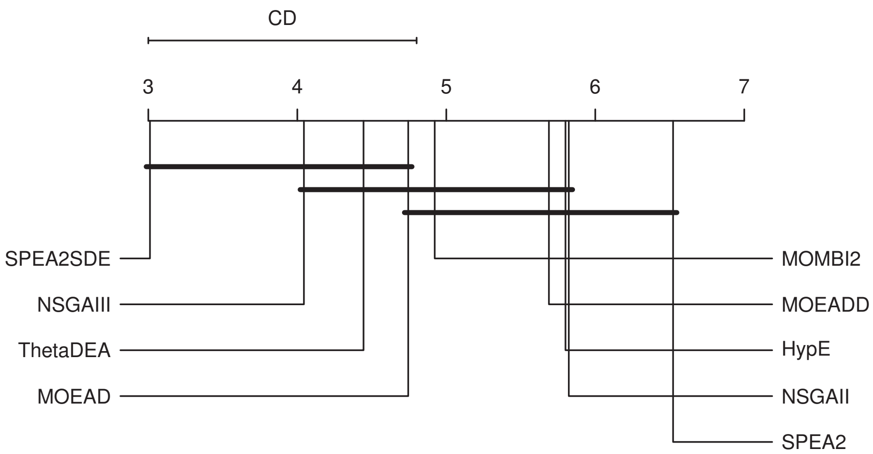


Figure C.2: Critical difference plot for the HV indicator. The algorithms connected by a bold horizontal line are considered statistically equivalent with 95% of significance.

The overall analysis of the nine algorithms by the IGD quality indicator (Figure C.1) demonstrated that no algorithm outperforms all others. In fact, for most of them, we did not find a statistically significant difference. In the critical difference plot, the second bold line connects seven of the nine MOEAs. Moreover, SPEA2SDE and NSGA-II (both using Pareto dominance as the first criteria) were the best overall performing MOEAs to the IGD quality indicator. In the per problem analysis (presented at Table C.1), SPEA2SDE and NSGA-II stand out mainly on MaF01 (inverted PF), MaF02 (concurrent convergence), and MaF13 (complicated variable linkages). After SPEA2SDE and NSGA-II, the best performing MOEAs are NSGA-III and ThetaDEA. They stand out mainly on MaF05, MaF10, and MaF12 (all biased, badly-scaled with triangular Pareto front shape). Next comes SPEA2 and MOEA/D, with good results on MaF08, MaF09 (both linear and degenerate), and MaF13 (concave, degenerate). Finally, we have MOEA/DD, MOMBI2, and HypE. MOEA/DD and MOMBI2 performed well on MaF14 (linear, partially separable, large scale, complicated fitness landscape). Hype performed well on MaF15 (convex, partially separable, large scale, inverted, complicated fitness landscape).

Table C.1: Average values for IGD comparing the nine MOEAs

Obj.	problem	HypE	MOEAD	MOEADD	MOMBI2	NSGAII	NSGAIII	SPEA2	SPEA2SDE	ThetaDEA
	MaF01	0.00216	0.00194	0.00234	0.00243	0.00160	0.00195	0.00134	**0.00116**	0.00238
	MaF02	0.0138	0.0110	0.00732	0.0106	0.00979	0.00909	**0.00679**	0.00712	0.00960
	MaF03	0.0573	0.0101	0.00160	0.00424	575	**0.000835**	3.73e+08	0.00124	0.00141
	MaF04	0.0318	0.0503	0.675	0.0327	0.0212	0.0362	**0.0188**	0.0381	0.0332
	MaF05	0.0862	0.0693	0.0593	**0.0225**	0.0252	0.0231	0.0243	0.0280	0.0231
	MaF06	0.00255	0.000432	0.000915	0.00371	4.03e-05	0.000594	**2.70e-05**	9.65e-05	0.00125
	MaF07	0.00573	0.00730	0.00604	0.00413	0.00333	0.00342	0.00316	**0.00313**	0.00348
5	MaF08	0.00991	0.00144	2.04	0.00400	0.00171	0.00318	**0.00114**	0.00130	0.00459
	MaF09	0.00784	0.00163	0.0111	0.00458	0.00717	0.00679	0.00122	**0.00105**	0.0106
	MaF10	0.0216	0.0246	0.0138	**0.00785**	0.00840	0.0107	0.0162	0.0112	0.00917
	MaF11	0.0920	0.0177	0.0599	0.0165	0.00958	0.0110	**0.00750**	0.0216	0.0132
	MaF12	0.0486	0.0325	0.0117	0.0118	0.0125	0.0106	0.0113	0.0123	**0.0106**
	MaF13	0.00397	0.00148	0.00193	0.00666	0.00168	0.00388	1.23e+03	**0.00112**	0.00360
	MaF14	0.00578	0.00785	**0.00410**	0.00696	0.274	0.0147	53.0	0.00474	0.0144
	MaF15	0.00671	**0.00325**	0.00513	0.00428	0.295	0.0126	0.0890	0.00366	0.0114
	MaF01	0.00425	0.00355	0.00613	0.00491	0.00352	0.00403	0.00345	**0.00270**	0.00415
	MaF02	0.00478	0.00375	0.00342	0.00623	**0.00204**	0.00266	0.00209	0.00215	0.00267
	MaF03	38.2	0.00218	0.00184	0.00617	4.23e+03	0.00569	1.70e+10	**0.00174**	0.00203
	MaF04	1.03	3.60	98.4	1.95	0.802	1.87	**0.772**	1.97	1.71
	MaF05	2.43	4.45	4.39	4.29	1.27	**1.18**	2.08	1.70	1.23
	MaF06	0.00251	**0.000284**	0.00211	0.00653	0.00399	0.00394	1.46	0.00563	0.00286
	MaF07	0.0565	0.00917	0.0148	0.0151	0.0119	0.0120	0.0177	**0.00749**	0.00827
10	MaF08	0.00588	0.00190	0.0227	0.0146	0.00210	0.00540	**0.00152**	0.00172	0.0110
	MaF09	0.0212	0.00343	0.0687	0.0143	0.440	0.0120	0.607	**0.00138**	0.0111
	MaF10	0.0426	0.0393	0.0344	0.0249	0.0219	0.0241	0.0364	0.0248	**0.0216**
	MaF11	0.129	0.101	0.114	0.0704	0.0160	0.0371	**0.0156**	0.0673	0.0332
	MaF12	0.0905	0.0855	0.0815	0.0653	0.0612	0.0573	0.0585	**0.0546**	0.0568
	MaF13	0.00397	0.00317	0.00435	0.00795	0.00175	0.00491	**0.00116**	0.00120	0.00669
	MaF14	1.46	0.00945	0.00895	0.00955	1.18	0.0454	361	**0.00447**	0.0450
	MaF15	0.0113	0.0557	0.0125	0.0127	1.29	0.0105	3.45	**0.00922**	0.0154
	MaF01	0.00732	0.00543	0.00765	0.00599	0.00489	0.00503	0.00519	**0.00449**	0.00493
	MaF02	0.00727	0.00520	0.00465	0.0107	**0.00251**	0.00392	0.00332	0.00330	0.00415
	MaF03	9.40e+05	0.00245	**0.00206**	0.00694	421	0.00741	2.28e+10	0.00212	0.00457
	MaF04	998	50.7	3.20e+03	104	33.6	69.1	**28.8**	95.8	83.4
	MaF05	98.6	125	117	125	**36.0**	55.0	76.7	50.7	55.0
	MaF06	0.242	**0.000335**	0.00198	0.00791	0.00462	0.00448	2.03	0.00440	0.00356
	MaF07	0.368	0.0152	0.0260	0.0520	0.0262	0.0684	0.0735	**0.0123**	0.0564
15	MaF08	0.00825	0.00244	7.90	0.0280	0.00356	0.00743	**0.00243**	0.00311	0.0154
	MaF09	0.0683	0.0102	0.0801	0.0678	0.0780	0.0171	0.00662	**0.00225**	0.0404
	MaF10	0.0578	0.0534	0.0506	0.0550	**0.0315**	0.0338	0.0546	0.0473	0.0334
	MaF11	0.219	0.205	0.205	0.201	**0.0159**	0.0604	0.0374	0.155	0.138
	MaF12	0.226	0.174	0.150	0.143	0.121	0.121	0.132	**0.119**	0.121
	MaF13	0.00528	0.00563	0.00607	0.0110	0.00453	0.00659	2.63e+04	**0.00179**	0.00801
	MaF14	0.0223	0.0121	0.00754	0.00934	0.297	0.0190	662	**0.00559**	0.0156
	MaF15	0.0140	0.110	0.0153	0.0196	0.852	0.0672	4.39	**0.0136**	0.0184

On the hypervolume analysis (Figure C.2 and Table C.2), the best performing MOEAs are SPEA2SDE, NSGA-III, and ThetaDEA, mainly on MaF04 (inverted), MaF05, and MaF12 (both badly-scaled triangular). Then come MOEA/D and MOMBI2, with good results on MaF06 (degenerate) and MaF10 (complicated mixed geometries). And finally, we have MOEA/DD (MaF02 and MaF14), HypE (MaF10), NSGA-II (MaF04), and SPEA2 (MaF08).

Table C.2: Average values for hypervolume comparing the nine MOEAs

Obj.	problem	HypE	MOEAD	MOEADD	MOMBI2	NSGAII	NSGAIII	SPEA2	SPEA2SDE	ThetaDEA
5	MaF01	0.00600	0.00634	0.00568	0.00582	0.00806	0.00693	0.00849	**0.0129**	0.00561
	MaF02	0.146	0.147	0.183	0.162	0.159	0.179	0.160	**0.204**	0.171
	MaF03	0.264	0.943	0.992	0.761	0.00	**0.999**	0.00	0.992	0.992
	MaF04	0.0655	0.0258	0.00	0.0450	0.0964	0.0640	0.0947	**0.106**	0.0772
	MaF05	0.504	0.538	0.683	0.809	0.629	0.812	0.702	0.774	**0.813**
	MaF06	0.102	0.120	0.0982	0.107	**0.130**	0.121	0.129	0.129	0.116
	MaF07	0.142	0.106	0.145	0.256	0.203	0.247	0.190	**0.270**	0.215
	MaF08	0.0584	0.119	0.0385	0.0855	0.115	0.0965	0.124	**0.126**	0.0802
	MaF09	0.172	0.315	0.108	0.227	0.182	0.189	0.317	**0.324**	0.132
	MaF10	**0.987**	0.750	0.960	0.970	0.917	0.923	0.427	0.947	0.930
	MaF11	0.984	0.984	0.970	0.993	0.989	**0.995**	0.977	0.984	0.995
	MaF12	0.285	0.494	0.733	0.761	0.615	0.763	0.598	0.744	**0.769**
	MaF13	0.205	0.276	0.244	0.105	0.236	0.120	0.240	**0.295**	0.147
	MaF14	0.535	0.273	0.595	0.525	0.00	0.0318	0.00	**0.745**	0.0936
	MaF15	0.00716	0.0271	0.0312	0.0489	0.00	1.18e-05	0.00	**0.103**	0.00166
10	MaF01	5.00e-08	0.00	0.00	3.00e-07	5.00e-08	**3.50e-07**	0.00	2.50e-07	2.50e-07
	MaF02	0.105	0.169	0.203	0.126	0.189	0.195	0.146	**0.226**	0.190
	MaF03	0.00	**1.00**	0.985	0.614	0.00	0.841	0.00	0.998	0.990
	MaF04	2.12e-05	1.00e-07	0.00	1.50e-07	2.74e-05	0.000147	8.60e-06	3.10e-06	**0.000206**
	MaF05	0.293	0.445	0.555	0.707	0.00	0.966	0.00	0.691	**0.968**
	MaF06	0.0964	**0.0982**	0.0105	0.0928	0.0606	0.0647	0.0150	0.0476	0.0551
	MaF07	0.0320	0.000222	0.000126	0.173	6.18e-05	0.167	3.50e-06	0.0235	**0.183**
	MaF08	0.00686	0.00938	0.00100	0.00409	0.00880	0.00745	0.0102	**0.0109**	0.00501
	MaF09	0.00262	0.0128	0.000300	0.00614	4.70e-05	0.00514	4.71e-05	**0.0184**	0.00562
	MaF10	**0.997**	0.950	0.898	0.971	0.728	0.903	0.240	0.938	0.949
	MaF11	0.998	**0.999**	0.956	0.994	0.998	0.995	0.927	0.992	0.992
	MaF12	0.302	0.580	0.690	0.857	0.574	0.864	0.515	0.844	**0.895**
	MaF13	0.125	0.109	0.0756	0.0523	0.120	0.0304	0.108	**0.141**	0.00576
	MaF14	0.360	0.355	0.460	0.433	0.00	0.00712	0.00	**0.960**	0.0203
	MaF15	0.00	0.00	5.00e-08	0.00	0.00	2.30e-06	0.00	**7.45e-05**	0.00
15	MaF01	0.00	0.00	**0.00**	0.00	0.00	0.00	0.00	0.00	0.00
	MaF02	0.0678	0.152	0.161	0.0714	0.116	0.158	0.0813	**0.209**	0.166
	MaF03	0.475	**1.00**	0.983	0.598	0.00	0.608	0.00	0.999	0.806
	MaF04	0.00	0.00	0.00	0.00	0.00	**3.50e-07**	0.00	0.00	5.00e-08
	MaF05	0.0962	0.355	0.404	0.534	0.00	0.991	0.00	0.697	**0.991**
	MaF06	0.0137	**0.0938**	0.0557	0.0897	0.0557	0.0697	0.00	0.0113	0.0910
	MaF07	0.00	5.00e-08	1.35e-06	0.0643	0.00	0.145	0.00	0.00257	**0.155**
	MaF08	0.000222	0.000514	0.000171	2.88e-05	0.000328	0.000255	0.000488	**0.000576**	0.000154
	MaF09	8.21e-05	0.000146	0.000119	4.55e-05	4.25e-05	0.000340	0.000480	**0.00112**	0.000229
	MaF10	0.988	**0.995**	0.883	0.950	0.956	0.964	0.203	0.934	0.954
	MaF11	0.994	0.999	0.953	0.984	**0.999**	0.998	0.888	0.992	0.825
	MaF12	0.206	0.463	0.551	0.790	0.491	0.901	0.320	0.841	**0.916**
	MaF13	0.0836	0.0672	0.0580	0.0231	0.0703	0.00524	0.0357	**0.0880**	0.000937
	MaF14	0.0256	0.222	0.572	0.521	0.00	0.00103	0.00	**0.905**	0.0181
	MaF15	0.00	0.00	**0.00**	0.00	0.00	0.00	0.00	0.00	0.00

Those observations demonstrate the presence of the No Free Lunch theorem on many-objective optimization. Also, it demonstrates the diversity of characteristics of the benchmark problem suite and the variety of algorithms used in the pool of low-level heuristics.